

BILKENT UNIVERSITY

CS492 - Senior Design Project II

Final Report

T2423 - Evalio

Eren Hayrettin Arım	22002306
Ahmet Reşat Demir	22002299
Mehmet Burak Demirel	22003396
Dilara Mandıracı	22101643
Yusuf Toraman	22002885

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1. Introduction	5
2. Requirements Details	5
2.1 Functional Requirements	5
2.1.1 Questions Database and Exam Builder:	5
2.1.2 Analysis Tools	6
2.1.3 Portfolio Builder	6
2.1.4 Exam Result Viewing Session Organizer	7
2.2 Nonfunctional Requirements	7
2.2.1 User Friendliness / Usability	7
2.2.2 Maintainability	7
2.2.3 Reliability	7
2.2.4 Security	8
3. Final Architecture and Design Details	8
3.1 Evalio Final Architecture	8
3.2 Overview	9
3.3 Subsystem Decomposition	. 11
3.4 Hardware/Software Mapping	. 11
3.5 Persistent Data Management	.12
3.6 Access Control and Security	. 12
4. Development/Implementation Details	. 13
4.1. Frontend	. 13
4.1.1. Layout & Navigation	. 14
4.1.2. Styling & Theme	. 15
4.1.3. Deployment	. 15
4.2. Backend	16
4.2.1. Microservice Setup	. 17
4.2.2. API Gateway	17
4.2.3. Service Discovery & Health	. 18
4.2.4 Security & Configuration	.18
4.2.5 Common Code Re-use	.19
4.2.6. Containerisation & Deployment	. 19
4.3 Storage	.20
4.3.1. Database Tables	.21
4.3.1.1. User Service Tables	.21
4.3.1.2. Question & Exam Service Tables	. 22
4.3.1.3. Slot Service Tables	. 24
4.3.1.4. Portfolio Service Tables	. 25
4.3.2. Google Cloud Storage	.27
5. Test Cases and Results	. 28
5.1 Functional Test Cases	28
5.1.1 Authentication Test Cases	.28

5.1.2 Question Database Test Cases	31
5.1.3 LaTeX Service Test Cases	. 34
5.1.4 Exam Creation Test Cases	37
5.1.5 Slot Test Cases	. 39
5.1.6 Portfolio Builder Test Cases	45
5.2 Non-Functional Test Cases	51
6. Maintenance Plan and Details	53
6.1. Monitoring & Health Checks	53
6.2. Scheduled Maintenance Tasks	54
6.3. Scaling & Capacity Planning	54
7. Other Project Elements	. 55
7.1. Consideration of Various Factors in Engineering Design	55
7.1.1 Constraints	55
7.1.1.1 Authentication & Data Access	55
7.1.1.2 Question Scanning	55
7.1.1.3 Cloud Services & Containerization	55
7.1.1.4 Advertisement	55
7.1.1.5 Time Management	56
7.2. Ethics and Professional Responsibilities	56
7.3. Teamwork Details	56
7.3.1. Contributing and functioning effectively on the team to establish goals, plan	
tasks, and meet objectives	56
7.3.2. Helping creating a collaborative and inclusive environment	57
7.3.3. Taking lead role and sharing leadership on the team	57
7.3.4. Meeting objectives	58
7.4 New Knowledge Acquired and Applied	60
8. Conclusion and Future Work	61
8.1 Conclusion	61
8.2 Future Work	61
9. User Manual	63
9.1 General	63
9.1.1 Landing Page	63
9.1.2 Login	64
9.1.3 Registration Request	65
9.2 Admin Management	66
9.3.1 Admin Page	66
9.3 Profile Pages	68
9.3.1 Institution Profile Page	. 68
9.3.2 Instructor Profile Page	72
9.3.3 Teaching Assistant Profile Page	75
9.4 Question & Exam Module	76
9.4.1 List Questions	76
9.4.2 Create Question	77
9.4.3 Show Question Details	78

9.4.4 Edit Question	79
9.4.5 Create New Exam	80
9.4.6 Add Question To Selected Exam	81
9.4.7 Show Exam Details	82
9.4.8 Edit Exam Details	83
9.5 Objection Module	84
9.5.1 Creating an Objection Session (TA/Instructor)	84
9.5.2 Managing the Objection Session (TA/Instructor)	86
9.5.3 Making Objection Session Reservations (Student)	89
9.6 Portfolio Module	92
9.6.1 Courses and Portfolio Page	92
9.6.2 Add New Course Modal	93
9.6.3 Filter Modal	94
9.6.4 Course Summary Page	95
9.6.5 Assign TA modal	97
9.6.6 Add Portfolio Item Modal	98
9.6.7 Portfolio Files Page	99
9.6.8 Portfolios Page/ Add File Modal	. 101
9.6.9 Portfolios Page/ Add Portfolio Type Modal	. 102
9.6.10 Portfolios Page/ Add Portfolio Type Modal	. 103
9.6.11 Portfolio Creation Page	104
10. Glossary	.106
11. References	. 109

1. Introduction

Even if the traditional paper-based exam method is widely used, it often falls short of accurately measuring student success and meeting modern educational needs. Research from Purdue University shows the importance of mapping and analyzing questions during the exam preparation phase to enhance exam quality and performance [1]. In response, our team aims to develop Evalio, an advanced exam database and management application designed to streamline exam creation and improve assessment effectiveness. Evalio enables instructors to create practical, high-quality exams through a shared, searchable question database that supports multiple courses. With features like question usage tracking, editing, versioning, export options, and detailed performance analytics, Evalio helps create balanced assessments while managing assignments and exams. Additionally, it offers secure, multi-university collaboration. It ultimately contributes to improve teaching and learning outcomes.

2. Requirements Details

2.1 Functional Requirements

The functional requirements of this project focus on creating a secure, efficient, and user-friendly system to manage question databases, build exams, analyze performance records, and organize exam result viewing sessions. Each module is designed to streamline workflows for instructors, teaching assistants, and students. The system also supports accreditation needs and inter-university collaboration.

2.1.1 Questions Database and Exam Builder:

Maintain a question database for each course, categorized by type and topic classification.

- Instructors can add new questions to the database.
- Log the history of question usage, including statistical data.
- Allow the addition, editing, and deprecation of questions.

- Implement permission controls to restrict or allow access to specific course-related question pools.
- Provide an exam builder that supports the following:
 - Question selection and arrangement.
 - Editable space and points allocation for questions.
 - Header section customization (course name, date, name/ID fields, score table).
 - Export to LaTeX, Google Docs, or other visual formatting editors, with optional PDF export.
 - In-editor question editing and creation.
- Enable instructors to build modular exams by selecting questions from the database using filters such as difficulty level, topic, and type.
- Include functionality to auto-generate balanced exams based on predefined criteria, such as equal topic distribution or desired difficulty levels.

2.1.2 Analysis Tools

- University and course instructors can create custom reports based on selected metrics:
 - Correlation between attendance and academic performance.
 - Success rate per question.
 - Success rate comparison with other universities per question.
 - etc....
- University and course instructors can create ABET/MÜDEK reports according to global standards.

2.1.3 Portfolio Builder

- Enable TAs to upload scans of selected exams (best, average, worst), quizzes, homework, projects, and exam seating plans.
- Upload attendance sheets and grading files.
- Generate course summary forms from uploaded and system data.
- Export a merged PDF containing all portfolio elements, including exam solution keys.

2.1.4 Exam Result Viewing Session Organizer

- Enable teaching assistants to create exam result viewing sessions.
- Enable teaching assistants to create time slots for created sessions, which allows configurable numbers.
- Allow teaching assistants to cancel or rearrange sessions.
- Allow users to check schedules via the institution's local system to avoid conflicts.
- Allow students to access the session with the provided session key and password.
- Announce and manage session sign-ups on a first-come, first-served basis.

2.2 Nonfunctional Requirements

2.2.1 User Friendliness / Usability

Evalio aims to be user-friendly with its easy and understandable interface, designed to facilitate users while creating exams by taking advantage of the modularity of the questions. We know that currently, instructors can create exams using different tools, so this product focuses on providing advanced ease of use. Therefore, the user interface should be as simple as possible while providing the functionality of the basic modules.

2.2.2 Maintainability

Since Evalio targets various types of end users, such as instructors, students, and teaching assistants, and includes different modules, we aim to make it as sustainable as possible by splitting the architecture into services. A modular codebase will enable isolated updates without breaking other services or modules.

2.2.3 Reliability

Evalio aims to be reliable. Given the variety of end-user types and use cases, a single point of failure should not affect other modules, and any failure should be

quickly recoverable. We aim to increase reliability by splitting the application's architecture into separate services.

2.2.4 Security

Since we aim to support different user types (e.g., instructors and institutions), the application should include a user authentication and authorization system to ensure that each user type has the correct permissions to access specific modules. Additionally, passwords and potentially sensitive information should be stored in a coded form in the database to ensure security.

3. Final Architecture and Design Details

3.1 Evalio Final Architecture



Figure 1: Final High-Level System Architecture Diagram

Evalio's architecture follows a microservices-based approach, ensuring modularity, scalability, and maintainability. It consists of a frontend, API gateway, microservices, a data layer, and shared utilities. Microservices handle exam creation, question

management, authentication, course portfolio management, and objections, operating independently for seamless updates and expansion. The data layer includes PostgreSQL for structured storage and Redis for caching, ensuring high performance. Consul supports service discovery, while TeXLive LaTeX Compiler enables LaTeX-based exam generation. Evalio also uses Evalio-Common, a shared library with reusable database models, authentication logic, and schema definitions, ensuring consistency across services. This modular design enables secure access control, efficient data handling, and streamlined exam management for universities and institutions.

3.2 Overview

Evalio follows a distributed microservices architecture, which separates core functionalities into independent services. The system consists of the following key components:

- Application Layer: A web-based front-end that provides users with an intuitive interface for exam creation, question management, and analytics visualization.
- **API Gateway**: A central entry point for all client requests, ensuring secure communication between services and managing request routing.
- Microservices Layer:
 - Exam/Solution Builder Service: Manages exam creation, LaTeX rendering, and question selection.
 - Question Service: Stores, retrieves, and manages exam questions with metadata (difficulty, success rate, versioning).
 - Authentication Service (Auth Service): Handles user authentication, role-based access control (RBAC), and session management.
 - Course Portfolio Builder Service: Enables the compilation of institutional course portfolios for accreditation.

- Objection Service: Organizes post-exam result viewing sessions and manages student objections.
- Analytics Service: Generates performance reports, success rate statistics, and accreditation compliance reports.
- Data Layer: Manages storage through:
 - PostgreSQL (relational database for structured exam and user data)
 - Redis (caching and real-time session management)
 - Google (cloud-based object storage for scanned exam documents and portfolio files)

• Infrastructure Layer:

- Consul [2] (for service discovery and health checks)
- TeXLive LaTeX Compiler (to generate LaTeX-based exam PDFs)

3.3 Subsystem Decomposition



Figure 2: Proposed Subsystem Decomposition Diagram

3.4 Hardware/Software Mapping

Evalio is a software-only system with no dedicated hardware components. It operates on cloud-based infrastructure, utilizing microservices, databases, and caching for scalability and reliability. No hardware-software mapping is required beyond standard server deployments.

3.5 Persistent Data Management

Evalio keeps its structured data in PostgreSQL and utilizes Redis for cache purposes, thus ensuring scalable and efficient data storage. The system also features GCS as an object storage for storing large documents in the Portfolio Builder Service.

To enhance database scalability and performance, we aim to partition large data sets with table partitioning and query optimization with indexes. GCS lifecycle policies can also manage storage expenses by automatically migrating older documents to lower-cost storage classes. With a responsive and cost-effective system, these solutions ensure effective data persistence, maintainability, and long-term scalability.

3.6 Access Control and Security

This application implements a strict role-based access control system. This is because it contains sensitive information, such as exam questions, making it a critical security concern to control who can access different application areas. We have integrated role-based access control and defined specific access rules.

- Registration can only be performed by the institution. Instructors or TAs cannot register; only the institution can add them to the system.
- When an institution signs in, it gains access only after admin approval.
- Questions are tagged as public or private. Public questions are visible to all registered users in the system, while private questions can only be accessed by individuals affiliated with the respective institution.
- Students cannot register or access the system. They can only access the application through a temporary token sent via a link, which allows them to select a time slot for an objection session.
- The admin is permitted to perform CRUD operations on all entities within the system. This includes modifying, adding, and deleting questions and reports, as well as adding, deleting, or blocking users from the system.
- Instructors are allowed to view all questions in the database.

- TAs can only see the exams assigned to them and are not privileged to know all the question databases.
- Only instructors are allowed to modify questions.

In the below figure, you can see the access scope of different user types:



Figure 3: User access scope diagram

4. Development/Implementation Details

4.1. Frontend

The Evalio user interface is developed using React 18 and is structured around a role-driven, component-based architecture that aligns with the system's microservices backend. The main goal of the frontend is to deliver a transparent, scalable, and responsive experience tailored to five distinct user roles: Teaching Assistant, Instructor, Institution, Student, and System Admin.

Each role accesses the platform through specific views and workflows, supported by route-based permission control and dynamic UI rendering. This structure allows features such as exam creation, objection scheduling, performance analytics, and portfolio management to be presented fluently.

React's modular structure made it easier for our team to develop the interfaces for different user roles in parallel without losing consistency in design or behavior. By sharing familiar layout and state management patterns, we were able to keep the experience uniform across the application. We also focused on keeping the frontend loosely connected to the backend, so changes in APIs wouldn't break the interface.

4.1.1. Layout & Navigation

This section describes the overall layout structure and navigation logic of the Evalio frontend, focusing on how the interface adapts dynamically based on user roles and session state.

- Top-level Shell: A persistent AppLayout component renders the navigation bar, collapsible side menu, and notification tray.
- Role-aware Menus: Menu items are computed from the JWT's role claim and reactively updated on refresh.
- ProtectedRoute / PublicRoute wrappers enforce authentication and silently refresh expired tokens through the Gateway.

Page / Route	Student	TA	Instructor	Institution Admin	System Admin
/login, /register	No	Yes	Yes	Yes	Yes
/exam/create	No	No	Yes	No	No
/exam/detail/:id	No	No	Yes	No	No
/slot/reserve	Yes	No	No	No	No
/slot/create	No	Yes	Yes	No	No
/portfolio/list	No	Yes	Yes	No	No
/portfolio/summary/:id	No	Yes	Yes	No	No
/analyze	No	Yes	Yes	Yes	No
/admin	No	No	No	No	Yes

Figure 4: Role Access Table

4.1.2. Styling & Theme

Evalio's visual identity is built on a consistent color palette and a shared layout structure. A centralized theme configuration ensures all pages have uniform spacing, typography, and color choices.

Shared layout components such as buttons, tables, and headers are reused across different modules, promoting visual consistency and maintainability.

4.1.3. Deployment

The Evalio frontend is deployed as a static web application served via Nginx within a Docker container. The production build is generated using React's build script, and the output files are placed in the Nginx serving directory (*/usr/share/nginx/html*) [2].

A custom nginx.conf handles client-side routing and proxies /api requests to the backend Gateway, ensuring frontend and backend integration. Deployment is orchestrated using Docker Compose.yaml, while a helper script (manager.sh) simplifies building and running the container with commands like ./manager.sh build.

Since all code is served statically, environment-specific configurations (e.g., API base URLs) must be resolved at build time or handled server-side. The setup is easily extendable for both development and production environments.



Figure 5: Frontend Container Deployment and API Proxy Flow

4.2. Backend

The Evalio backend is organised around a microservice architecture that decomposes each business capability into an independently deployable and maintainable service. This architectural style was chosen to enable modular development, improve scalability, and allow different people to work on separate services in parallel without creating interdependencies. Each microservice encapsulates a specific domain logic and communicates with others through well-defined RESTful APIs.

This section outlines the core development choices, implementation strategies, and the supporting infrastructure that ensure Evalio remains scalable, secure, and adaptable to institutional requirements over time.



Figure 6: Evalio Backend High-Level Service Topology

4.2.1. Microservice Setup

Microservice Setup

Service	Core Responsibility
auth-service	OAuth 2.0 login, JWT issuance, RBAC enforcement
question-service	CRUD & metadata for version-controlled question bank
exam-service	Exam assembly, question selection, LaTeX render trigger
portfolio-service	Course portfolio CRUD & PDF export
slot-service	Result-view/objection slot scheduling
evalio-common	Shared data models, utilities and schemas for all services
scripts	Development and deployment helper scripts

Figure 7: Microservice Setup of Evalio

All services expose RESTful endpoints (/api/<service>/...) and are packaged as Docker images published to an internal registry. Evalio-common and scripts are the helper folders in the Evalio structure.

4.2.2. API Gateway

A lightweight FastAPI-based Gateway validates JWTs, can apply rate limits, and forwards traffic to the appropriate microservice. By centralizing cross-cutting concerns such as authentication, routing, and request handling, the Gateway keeps individual services thin and language-agnostic. NGINX is used as an API Gateway in the Evalio Backend.

This design choice also simplifies CORS handling, enforces a single public entry point, and aligns with zero-trust architecture principles. It additionally opens the door for future enhancements like request-level metrics, load-aware routing, and A/B testing at the API level.

4.2.3. Service Discovery & Health

All containers register with Hashicorp Consul at start-up. Consul performs:

- Active health checks (HTTP 200 probes every 10s), enabling auto-deregistration of unhealthy tasks.
- DNS-style discovery, so the gateway routes are by logical name (question-service.service). Currently, this feature of Consul is not actively used, but it can be easily added to facilitate communication and management.

4.2.4 Security & Configuration

Evalio's authentication infrastructure is based on JSON Web Tokens (JWT) [4] issued by the auth-service. These tokens include standard claims such as sub (subject), role, and exp (expiration), and are signed using an RSA private key. Other services validate the authenticity of incoming requests by verifying the JWT signature using the corresponding public key, ensuring secure and stateless access control across the system.

All sensitive configuration values, such as database credentials, service keys, and signing secrets, are managed through .env files loaded into each container at runtime. These environment files are mounted via Docker Compose and are never embedded into the container images, reducing the risk of credential leakage.

Internal communication between microservices occurs over a private Docker overlay network, isolating service traffic from the public internet and ensuring that all inter-service communication remains secure within the deployment environment.

In addition, extra user-side security systems are used in Evalio. On each login, an OTP code is sent to the user. The Admin approves institution registration, and the institution handles instructor registration. Students cannot register for the system, but can use it temporarily in the slot service.

4.2.5 Common Code Re-use

A dedicated evalio-common/ package holds

- Pydantic DTOs are shared between client-server and service-to-service.
- Error/success response schema helpers.
- Database models.
- Dependencies.
- Logging & tracing middleware.

Publishing this package to the local PyPI mirror ensures one-line upgrades and eliminates copy-paste across repos.

4.2.6. Containerisation & Deployment

Evalio is fully containerized using Docker, allowing for consistent local development and production environments. Each microservice, including the gateway, domain services, and utility tool, is built and packaged as a separate container image. This modular structure enables each service's independent development, testing, and deployment.

Service orchestration is managed using Docker Compose, separated into two main configurations:

- docker-compose.yaml defines core services such as auth-service, exam-service, portfolio-service, slot-service, and the gateway.
- docker-compose-infra.yaml handles supporting components like Consul, which provides service discovery and health checks.

A helper script named manager. It simplifies local development and testing by wrapping standard commands like build, up, and logs, enabling fast service bootstrapping during iterative development cycles.

Each container reads its .env file, separating secrets and environment-specific configurations from code to securely inject credentials and runtime variables. These values are mounted dynamically and are never baked into the images themselves, helping to enforce secure deployment practices.

All containers communicate over a private Docker overlay network, isolating internal service traffic from external exposure and ensuring secure microservice communication.

Finally, the whole system is deployed to a virtual server on AWS EC2, using Docker Compose directly on the instance to launch and manage services. This setup allows the platform to operate in a production-like environment while remaining easily portable to future orchestrators such as ECS or Kubernetes.



Figure 8: Container Architecture of Evalio

4.3 Storage

Evalio uses a centralized PostgreSQL database to store all structured application data, including users, questions, exams, portfolios, objection slots, and related metadata. Although the system adopts a microservice architecture, a shared relational database is used to simplify data management. Each microservice is responsible for interacting with its subset of database tables through well-defined ORM layers, without violating the encapsulation of domain logic.

This unified schema approach enables more straightforward implementation of relationships, such as assigning teaching assistants to exams or tracking which institution owns a particular question.

Alembic manages schema evolution in a controlled and reproducible way for database migrations. With Alembic, each structural change to the database (e.g., adding a column, modifying constraints) is versioned and applied consistently across all environments, ensuring that the database remains in sync with the application logic over time.

ORM models are defined using SQLAlchemy with declarative mappings, and the data layer is structured to support scalability. For clarity, common fields such as created_at, updated_at, and deleted_at have been intentionally omitted from the documentation tables as they are often automatically managed and not directly relevant to domain logic.

4.3.1. Database Tables

4.3.1.1. User Service Tables

The table below is used to store instructors and teaching assistants. Basic and necessary user information is stored in the table. These two tables hold the institution_id in addition to the institution table, because it is essential to know which institution the TA and the instructor are affiliated with.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
name	varchar	
email	varchar	UNIQUE
password	varchar	
role	varchar	
last_login_at	timestamp	
blocked	boolean	
blocked_at	timestamp	
institution_id	varchar	FOREIGN KEY (institution)

Figure 9: Instructor and Teaching Assistant Table

The table below is used to store institutions. Basic and necessary institution information is stored in the table.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
name	varchar	
email	varchar	UNIQUE
password	varchar	
role	varchar	
last_login_at	timestamp	
blocked	boolean	
blocked_at	timestamp	

Figure 10: Institution Table

4.3.1.2. Question & Exam Service Tables

The table below shows the Question table. A question may have an exam, or the question may have unique properties. For this reason, there are fields such as is_private and required_space.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
parent_id	varchar	FOREIGN KEY (question)
title	varchar	
tags	varchar	
difficulty	varchar	
success_rate	float	
is_private	boolean	
institution_id	varchar	FOREIGN KEY (institution)
latex_file	varchar	
required_space	varchar	
description	varchar	
instructor_id	varchar	FOREIGN KEY (instructor)
is_favorite	boolean	

Figure 11: Question Table

The Exam Question table is similar to the Question table above, but this time it stores information about the Question related to a specific Exam.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
parent_id	varchar	FOREIGN KEY (question)
question_number	int	
title	varchar	
tags	varchar	
grade	float	
assigned_grade	float	
latex_file	varchar	
required_space	varchar	
description	varchar	
exam_id	varchar	FOREIGN KEY (exam)
institution_id	varchar	FOREIGN KEY (institution)
instructor_id	varchar	FOREIGN KEY (instructor)

Figure 12: Exam Question Table

The Exam table is similar to the Question table above, but it stores information about the Exam this time.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
title	varchar	
description	varchar	
exam_instructions	varchar	
status	varchar	
date	timestamp	
latex_file	varchar	
duration	float	
instructor_id	varchar	FOREIGN KEY (instructor)
course_id	varchar	FOREIGN KEY (course)

Figure 13: Exam Table

This table holds the relation between the Exam and the TA

Column Name	Data Type	Constraints
exam_id	varchar	FOREIGN KEY (exam), PRIMARY KEY
ta_id	varchar	FOREIGN KEY (teaching_assistant), PRIMARY KEY

Figure 14: Exam TA Association Table

4.3.1.3. Slot Service Tables

The table below contains basic information about the objection session. There is a Slot and a Day Range in an Objection Session.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
created_by	varchar	
slot_duration_min	int	
slot_capacity	int	
session_password	varchar	
class_name	varchar	
title	varchar	
status	varchar	

Figure 15: Objection Session Table

The Objection Day Range table keeps the day ranges of an Objection. An Objection can be scheduled over multiple days.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
session_id	varchar	FOREIGN KEY (objection_sessions)
start_datetime	datetime	
end_datetime	datetime	

Figure 16: Objection Day Range Table

The Slot table represents a single Slot within an Objection. A Slot may have more than one capacity, and its status may vary, such as disabled or active, so it also has fields like this.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
session_id	varchar	FOREIGN KEY (objection_sessions)
start_time	datetime	
end_time	datetime	
capacity	int	
status	varchar	

Figure 17: Slot Table

The reservation table lets you know the status of a particular slot within a specific objection session. In this table, the student-slot relationship can be ensured.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
slot_id	varchar	FOREIGN KEY (slots)
student_name	varchar	
status	varchar	

Figure 18: Reservation Table

4.3.1.4. Portfolio Service Tables

This table holds static information about university courses such as name, code, department, and description. It is linked to an institution.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
name	varchar	
code	varchar	
department	varchar	
description	varchar	
institution_id	varchar	FOREIGN KEY (institution)

Figure 19: Course Table

The Course Semester table stores each course's temporal and academic term-specific details, including year, semester, instructor, and number of sections.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
year	varchar	
semester	varchar	
number_of_sections	int	
course_id	varchar	FOREIGN KEY (course)
instructor_id	varchar	FOREIGN KEY (instructor)
is_current	boolean	

Figure 20: Course Semester Table

This table tracks uploaded portfolio documents for each instructor. It links to the portfolio type, instructor, and optionally a portfolio list.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
title	varchar	
description	varchar	
section	int	
status	varchar	
portfolio_file	varchar	
instructor_id	varchar	FOREIGN KEY (instructor)
portfolio_type_id	varchar	FOREIGN KEY (portfolio_type)
portfolio_list_id	varchar	FOREIGN KEY (portfolio_list)
order	int	

Figure 21: Portfolio Table

The portfolio type table defines metadata for different portfolio categories, such as exams or quizzes, including submission count, score averages, and assigned dates.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
type_name	varchar	
main_type	varchar	
description	varchar	
number_of_submissions	int	
assigned_date	datetime	
average_score	int	
is_assignment	varchar	
course_semester_id	varchar	FOREIGN KEY (course_semester)

Figure 22: Portfolio Type Table

The table below groups portfolios under a single list for bulk export or accreditation purposes. Summary and merged file fields support PDF generation.

Column Name	Data Type	Constraints
id	varchar	PRIMARY KEY
title	varchar	
course_semester_id	varchar	FOREIGN KEY (course_semester)
summaryHead	varchar	
summaryFooter	varchar	
merged_file	varchar	

Figure 23: Portfolio List Table

4.3.2. Google Cloud Storage

Evalio uses Google Cloud Storage (GCS) to store large unstructured files, particularly within the Portfolio Service, where instructors upload final portfolio documents in PDF format. These files, which can include multiple exam artifacts and accreditation materials, are stored as objects in GCS buckets to ensure durability and scalability beyond the limits of traditional relational databases.

Instead of embedding large binaries in PostgreSQL, Evalio offloads them to GCS and stores only the corresponding URLs or references in the database. This approach significantly reduces database load, improves file retrieval performance, and enables seamless integration with PDF generation and merging workflows. Access to uploaded files is secured through signed URLs, ensuring only authorized users can access the content during export or download operations.

5. Test Cases and Results

5.1 Functional Test Cases

5.1.1 Authentication Test Cases

Test ID:	FT001	Category:	Authentication	Severity:	Major	
Objective	Verify that an institution can successfully register with the system.					
Steps	 Open the Click on t Fill out th Click the Once the go to the en 	application (F he register bu e necessary in register button admin approvinail address in	Role: Institution). tton. nformation on the r n to complete the r ves the registration n the registration fo	registration for registration re n request (at prm.	orm. equest. most in 2 days),	
Expected result	The system successfully registers a verified institution.					
Date / Result	01/05/2025	- Pass				

Test ID:	FT002	Category:	Authentication	Severity:	Critical	
Objective	Verify that other user types, rather than institutions, cannot register for the system.					
Steps	 Open the Click on t Fill out th Click the Once the The user 	 Open the application (Role: TA, Instructor, Student, or any other user). Click on the register button. Fill out the necessary information on the registration form. Click the register button to complete the registration request. Once the admin receives the request, it will reject registration. The user will get an email about an unauthorized registration request. 				
Expected result	The system successfully rejects any other user's registration request.					
Date / Result	01/05/2025	- Pass				

Test ID:	FT003	Category:	Authentication	Severity:	Critical	
Objective	Verify that u	Verify that unregistered/unverified users cannot log in.				
Steps	 User click Fill out th User gets 	 User clicks login. Fill out the requested information on the login form. User gets a message 'there is no registered user with this email.' 				
Expected result	The system successfully prevents unregistered/unverified people from logging into the application.					
Date / Result	01/05/2025	- Pass				

Test ID:	FT004	Category:	Authentication	Severity:	Critical		
Objective	Verify that re	Verify that registered/verified users can log in.					
Steps	 Click logi Fill out th The user Check the Return to Click the You logge 	 Click login. Fill out the requested information on the login form. The user gets a message: 'Your 2fa code is sent to your email.' Check the email and copy the OTP code. Return to the login page and enter the OTP to verify 2FA. Click the button to continue You logged in. 					
Expected result	The system authentication	successfully a	allows registered ι	isers to log ir	ו with 2-factor		
Date / Result	01/05/2025	- Pass					

Test ID:	FT005	Category:	Authentication	Severity:	Critical			
Objective	Verify that an institution can register multiple TAs/Instructors to the system.							
Steps	 Log in as Go to the Click on t Click "cho Click "cho Choose t "name, ema Click Cho Click the 	an institution institution pro he register TA pose .csv file." he .csv file sa ill" of the respo pose. register buttor	ofile page. A or Instructor butto wed to the compute ective people.	on er and alread	ly filled with the			
Expected result	The system sends emai	The system successfully registers respective users to the system and sends emails to the users with their temporary passwords.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT006	Category:	Authentication	Severity:	Critical			
Objective	Verify that a system.	Verify that an institution can register a single TA/Instructors to the system.						
Steps	 Log in as Go to the Click on t Fill out th Click the 	 Log in as an institution. Go to the institution profile page. Click on the register TA or Instructor button Fill out the form with the necessary information Click the register button. 						
Expected result	The system successfully registers the respective user with the system and sends an email with their temporary password to the user.							
Date / Result	01/05/2025	- Pass						

5.1.2 Question Database Test Cases

Test ID:	FT007	Category:	Question DB	Severity:	Major				
Objective	Verify that a	Verify that an instructor can create and add a question to the database.							
Steps	 Log in as Navigate Click on t Enter vali Click "Su 	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Enter valid question details. Click "Submit." 							
Expected result	The system displays a c	The system successfully saves the question in the database and displays a confirmation message.							
Date / Result	01/05/2025	- Pass							

Test ID:	FT008	Category:	Question DB	Severity:	Major		
Objective	Ensure the system prevents question creation if required fields are missing.						
Steps	 Log in as Navigate Click on t Leave on required spa Click "Su 	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Leave one or more blank question fields (e.g., question text or required space). Click "Submit." 					
Expected result	The system missing. The	displays an e e question is r	rror message indic not saved.	cating the rec	uired fields are		
Date / Result	01/05/2025	- Pass					

Test ID:	FT009	Category:	Question DB	Severity:	Major			
Objective	Ensure an instructor can edit and update an existing question in the database.							
Steps	 Log in as Navigate Select an Click on t Modify th LaTeX conte Click "Sa 	an instructor. to the Question existing questi he "Edit Questi e question text ent with valid va ve Changes."	n List Page. ion from the list a ion" button. , metadata (tags, alues.	nd click that i type, difficult	row. y level), or			
Expected result	The system changes. A	successfully u confirmation m	pdates the questi essage is display	on and reflec ved after savi	ts the ng.			
Date / Result	01/05/2025	- Pass						

Test ID:	FT010	Category:	Question DB	Severity:	Minor			
Objective	Ensure that limit.	Ensure that the question description field has a reasonable character imit.						
Steps	 Log in as Navigate Click on t Enter a v allowed length 	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Enter a very long question description exceeding the maximum allowed length of 500. 						
Expected result	The system	The system prevents writing more than 500 characters and stops.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT011	Category:	Question DB	Severity:	Major			
Objective	Ensure the system handles question search queries with partial keywords appropriately.							
Steps	 Log in as Navigate Enter a p Click the Review th 	 Log in as an instructor. Navigate to the Question List Page. Enter a partial keyword (e.g., "inte" instead of "integral"). Click the "Search" button or enter the key. Review the displayed results. 						
Expected result	The system	The system should return related results.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT012	Category:	Question DB	Severity:	Major			
Objective	Ensure that (Difficulty Le	Ensure that instructors can simultaneously apply multiple filters (Difficulty Level + Tag + Institution) for questions.						
Steps	 Log in as Navigate Select minimization: E Review the 	 Log in as an instructor. Navigate to the Question List Page. Select multiple filters (Difficulty Level: Easy, Tags: Binary Search, Institution: Bilkent University). Review the displayed results. 						
Expected result	The system	The system displays only questions that meet all selected filter criteria.						
Date / Result	01/05/2025	- Pass						

5.1.3 LaTeX Service Test Cases

Test ID:	FT013	Category:	LaTeX	Severity:	Moderate			
Objective	Ensure that	Ensure that LaTeX syntax is handled correctly in the question text.						
Steps	 Log in as Navigate Click on t Open the Enter a q Wait for a 	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Open the LaTeX editor on that page. Enter a question with the valid LaTeX syntax. Wait for a second for automatic rendering. 						
Expected result	The system PDF withou	The system correctly processes and shows the converted LaTeX as a PDF without errors on the right side of the page.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT014	Category:	LaTeX	Severity:	Moderate			
Objective	Ensure that	Ensure that incorrect LaTeX syntax is handled in the question text.						
Steps	 Log in as Navigate Click on t Open the Enter a q Wait for a 	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Open the LaTeX editor on that page. Enter a question with the invalid LaTeX syntax. Wait for a second for automatic rendering. 						
Expected result	The system error."	Fhe system does not convert LaTeX to a PDF, showing "LaTeX syntax error."						
Date / Result	01/05/2025	- Pass						

Test ID:	FT015	Category:	LaTeX	Severity:	Moderate			
Objective	Ensure the system allows users to import a LaTeX file, validate its content, and display it correctly in the LaTeX editor.							
Steps	 Log in as Navigate Click on t Click on t Click on t Select a and upload Wait for a 	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Click on the "Import LaTeX File" option. Select a .tex file with a valid LaTeX syntax from the local system and upload it. Wait for a second for automatic rendering. 						
Expected result	The system file on the le the right sid	The system successfully uploads and displays the content of the .tex file on the left side. Also, the preview renders correctly as a PDF on the right side.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT016	Category:	LaTeX	Severity:	Moderate		
Objective	Ensure the system prevents users from importing LaTeX files with incorrect formatting.						
Steps	 Log in as an instructor. Navigate to the Profile Page. Click on the "Create New Question" button. Click on the "Import LaTeX File" option. Attempt to upload a corrupted or incorrectly formatted .tex file. Wait for a second for automatic rendering. 						
Expected result Date / Result	The system shows the LaTeX content in the editor but does not display the rendered PDF. The system shows a validation error: "LaTeX syntax error." 01/05/2025 - Pass						

Test ID:	FT017	Category:	LaTeX	Severity:	Critical		
Objective	Ensure that the system prevents users from importing non-LaTeX files.						
Steps	 Log in as an instructor. Navigate to the Question List Page. Click on "Add Question". Click on the "Import LaTeX File" option. Attempt to upload an unsupported file format (e.g., .docx, .pdf, .txt). 						
Expected result	The system will not list the unsupported file formats, and the user cannot select these files.						
Date / Result	01/05/2025 - Pass						

Test ID:	FT018	Category:	LaTeX/PDF	Severity:	Critical		
Objective	Ensure that the system allows instructors to export a question as a LaTeX or PDF file and that the exported content is correctly formatted.						
Steps	 Log in as an instructor. Navigate to the Question List Page. Select an existing question from the list. Click the "Export PDF" button. Click the "Export LaTeX" button. 						
Expected result	The system generates an adequately formatted LaTeX (.tex) and PDF (.pdf) file.						
Date / Result	01/05/2025 - Pass						
5.1.4 Exam Creation Test Cases

Test ID:	FT019	Category:	Exam	Severity:	Critical				
Objective	Ensure an i configuring	Ensure an instructor can build an exam by selecting questions and configuring exam settings.							
Steps	 Log in as Navigate Click the Enter the Enter the Click on t Wait for t Navigate Select a Select an button. Click on t Navigate Navigate 	an instructor to the Profile p "Create New E valid values fo he "Create" but he "Exam is sur to the Question question and cl he selected exa other question he selected exa to the Profile nd navigate the the page.	age. xam" button. r the exam field tton. ccessfully creat n List page ick the "Add Qu am on the oper and click the "A am on the oper page. e created exam	ds. ted" message. uestion to Exam ned exams list. Add Question to ned exams list. template in the	ı" button. o Exam"				
Expected result	The system questions. <i>A</i> questions o	shows exam p Also indicates th n the right side	arameters on t ne created PDF	he right side wi ⁻ with the comb	th added ination of				
Date / Result	01/05/2025	- Pass							

Test ID:	FT020	Category:	LaTeX, Exam	Severity:	Major
Objective	Ensure an i	nstructor can e	dit an existing e>	kam, modify its	s LaTeX
	content, and	a save the char	iges.		
Steps	1. Log in as	an instructor			
	2. Navigate	to the Profile p	age.		
	3. Select an	existing exam	template from the	ne exam list.	
	4. Click on t	he "Edit Exam"	button.		
	5. Change t	he order of the	questions by us	ing the drag a	nd drop
	feature, and	change the ex	am metadata.		
	6. Wait for a	second for au	tomatic renderin	g.	
	7. Click on t	he "Save Chan	ges" button.		
Expected result	The exam u	pdates succes	sfully, and a con	firmation mes	sage is
	displayed. T	he updated La	TeX content is c	orrectly rende	red and
	displayed.				
Date / Result	01/05/2025	- Pass			

Test ID:	FT021	Category:	Exam	Severity:	Critical				
Objective	Ensure that	Ensure that an instructor can export an exam as a PDF or LaTeX file.							
Steps	 Log in as Navigate Select an Click the Click the 	 Log in as an instructor Navigate to the Profile page. Select an existing exam template from the exam list. Click the "Export Exam LaTeX" button. Click the "Export Exam PDF" button. 							
Expected result	The system exported file	The system successfully generates a LaTeX and PDF file. The exported files contain all exam details formatted correctly.							
Date / Result	01/05/2025	- Pass							

5.1.5 Slot Test Cases

Test ID:	FT022	Category:	Slot	Severity:	Major				
Objective	Verify that a valid param	Verify that a TA/Instructor can create a new objection session using valid parameters.							
Steps	 Log in as Navigate Fill in all in slotDuration Click "Creation" 	 Log in as a TA/Instructor (You will be directed to the Profile page). Navigate to the "Create Objection Session" page. Fill in all required fields with valid data (startDatetime, endDatetime, slotDurationMin, etc.). Click "Create" to create a new objection session. 							
Expected result	The system created obje session stat	responds with ection session o us is set to AC	a success mes details (includin TIVE.	sage and retur g generated slo	ns the newly ots). The				
Date / Result	01/05/2025	- Pass							

Test ID:	FT023	Category:	Slot	Severity:	Major			
Objective	Verify that a more slots i	Verify that a TA/Instructor who owns the session can disable one or more slots in an ongoing objection session.						
Steps	1. Log in as 2. Click an <i>i</i> bar of the ta 3. Click one	 Log in as a TA/Instructor. Click an ACTIVE objection session from the "Created Objections" bar of the table. Click one of the ACTIVE slots to disable. 						
Expected result	The specific no longer re message.	ed slots change eserve them. Th	e status to "DIS/ ne system confi	ABLED," and st	udents can cess			
Date / Result	01/05/2025	- Pass						

Test ID:	FT024	Category:	Slot	Severity:	Major			
Objective	Ensure the end time be	Ensure the system appropriately handles invalid date/time inputs (e.g., end time before start time).						
Steps	 Log in as Navigate Enter a s 2025-03-0 Click Cre 	 Log in as a TA/Instructor. Navigate to "Create Objection Session." Enter a startDatetime that is later than the endDatetime (e.g., start = 2025-03-07 14:00, end = 2025-03-05 13:00). Click Create Session. 						
Expected result	The system "Invalid time	The system rejects the request, returning an error message such as "Invalid time range." No session is created.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT025	Category:	Slot	Severity:	Major			
Objective	Verify that a	student can su	uccessfully rese	erve an availabl	le slot.			
Steps	 A TA/Inst multiple slot Access th Select an Click Res Confirm F 	 A TA/Instructor has already created an objection session with multiple slots. Access the link to the objection session (enter their email). Select an available slot from the session. Click Reserve Slot. Confirm Reservation. 						
Expected result	The slot is r students ca (e.g., "Slot r	The slot is marked as reserved under the student's name, and other students can no longer select that same slot. A confirmation message (e.g., "Slot reserved successfully") appears.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT026	Category:	Slot	Severity:	Moderate				
Objective	Confirm tha taken by an	Confirm that a student cannot overwrite or reserve a slot already taken by another student.							
Steps	 A TA/Inst Access the if set). Another set Another set Navigate than the use Attempt to the the transmission of transmission of the transmission of transmission of the transmission of transmission of the transmission of the transmission of t	ructor has alreated the link to the ob- student has alreated to the same ob- er who already to reserve the a	ady created an ojection session eady reserved a ojection session reserved a part lready reserved	objection sessi (sessionPassy a particular slot. page as a diffe ticular slot. d slot.	on. word required				
Expected result	The system selection. A already rese	disables or gra n alert or warni erved").	ays out the take ng message ap	n slot and prev opears (e.g., "T	ents his slot is				
Date / Result	01/05/2025	- Pass							

Test ID:	FT027	Category:	Slot	Severity:	Moderate			
Objective	Verify that a further slot i	Verify that a TA/Instructor can close an objection session, preventing further slot reservations or modifications.						
Steps	 Log in as Navigate Select an Click the Confirm t 	 Log in as a TA/Instructor. Navigate to the existing objection session list. Select an ACTIVE session. Click the "Terminate Session" button. Confirm the status change. 						
Expected result	The objection no longer re remain visib	on session statu serve, modify, lle for reference	us changes to (or leave slots. I e.	CLOSED, and s Previously rese	students can rved slots			
Date / Result	01/05/2025	- Pass						

Test ID:	FT028	Category:	Slot	Severity:	Moderate				
Objective	Verify that a permanently	Verify that a TA/Instructor can delete an objection session, permanently removing all associated slots and reservations.							
Steps	 Log in as Navigate Click the terminated s Confirm t 	 Log in as a TA/Instructor. Navigate to the "created objections" table. Click the Trash Can button, which appears only next to the terminated sessions. Confirm the deletion. 							
Expected result	The system data. Stude	The system permanently deletes the session and all associated slot data. Students no longer see the session in the link.							
Date / Result	01/05/2025	- Pass							

Test ID:	FT029	Category:	Slot	Severity:	Critical			
Objective	Verify that a student cannot reserve a slot that the TA/Instructor has disabled.							
Steps	 A TA/Inst Access th (sessionPasis) Select a d Click to rest 	 A TA/Instructor creates an objection session. Access the link to the objection session as a Student (sessionPassword required if set). Select a disabled slot from the session. Click to reserve the slot. 						
Expected result	The system error messa	The system blocks the selection of the disabled slot and displays an error message like: "This slot is disabled."						
Date / Result	01/05/2025	- Pass						

Test ID:	FT030	Category:	Slot	Severity:	Major				
Objective	Verify that a	Verify that a slot cannot accept reservations beyond its capacity.							
Steps	 A TA/Inst . Three stu A fourth s Click Res 	 A TA/Instructor creates an objection session with a slot capacity = A TA/Instructor creates an objection session with a slot capacity = Three students reserve the slot. A fourth student tries to reserve the same slot. Click Reserve Slot. 							
Expected result	The system blocks the reservation and displays an error message: "This slot is full. Please select another available slot."								
Date / Result	01/05/2025	- Pass							

Test ID:	FT031	Category:	Slot	Severity:	Major		
Objective	Verify that a available or	Verify that a student can change their reserved slot to another available one.					
Steps	 A student The stude The stude The stude reserved slo Confirm r The stude Confirms the Click Res 	t enters the ses ent wants to ch ent cancels the ot. eservation can ent selects ano e change. eerve Slot.	sion and reserving oose a different in previous reserving reserving the state of th	ves a slot at 10 t time and tries ervation by click lot (e.g., 11:00	:00 AM. to change it. ing on a AM) and		
Expected result	The system releases the	successfully u e old one for otl	pdates the resence ner students.	ervation to the r	iew slot and		
Date / Result	01/05/2025	- Pass					

Test ID:	FT032	Category:	Slot	Severity:	Moderate			
Objective	Verify that a	Verify that a TA/Instructor can see the details of any objection session.						
Steps	 Log in as Navigate Select an Click on a 	 Log in as a TA/Instructor. Navigate to the created objection table. Select an objection session. Click on any slot to see reservations. 						
Expected result	The system containing s	The system successfully generates and downloads a PDF file containing session details.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT033	Category:	Slot	Severity:	Major		
Objective	Verify that n simultaneou issues.	Verify that multiple students attempting to reserve the same slot simultaneously do not cause inconsistencies or double-booking issues.					
Steps	 A TA/Inst capacity = 1 Multiple s reserve the 	 A TA/Instructor creates an objection session with multiple slots and capacity = 1. Multiple students (e.g., three students) simultaneously attempt to reserve the same available slot using parallel requests (JMeter). 					
Expected result	The system No duplicate The databas reservations	The system allows only one student to reserve the slot successfully. No duplicate bookings should be created in the database. The database maintains data integrity and does not allow conflicting reservations.					
Date / Result	01/05/2025	- Pass					

5.1.6 Portfolio Builder Test Cases

Test ID:	FT034	Category:	Portfolio	Severity:	Moderate			
Objective	Verify that th	Verify that the system can create a new course using valid parameters.						
Steps	 Log in as Navigate Click the Fill all the the opened Click the 	 Log in as an Instructor. Navigate to the portfolio main page Click the "Add New Course" button Fill all the required fields with valid data (course code, semester) in the opened modal. Click the "Create" button to submit. 						
Expected result	The system newly creat	The system responds with a success message and navigates to the newly created course page.						
Date / Result	01/05/2025	- Pass						

Test ID:	FT035	Category:	Portfolio	Severity:	Major		
Objective	Verify that th	Verify that the user can import a valid document type into the system.					
Steps	 Log in as Navigate Click the Click the Click the Fill all the (assessmer Click the Click the Select a Wait for a 	an instructor. to the portfolio course where "Add File" but required field at date). "Upload a File pdf or a .tex file second for a	o main page. the document will ton Is (type, subtype) a e" button. ile from the local sy utomatic rendering	be imported and optional ystem and up	fields bload it.		
Expected result	The system created file	responds witl in the portfolic	n a success messa o list.	age and lists	the newly		
Date / Result	01/05/2025	- Pass					

Test ID:	FT036	Category:	Portfolio	Severity:	Major			
Objective	Ensure that non-PDF file	Ensure that the system prevents users from importing non-Latex or non-PDF files.						
Steps	 Log in as Navigate Click the Click the Click the Fill all the (assessmer Click the Click the Attempt t Wait for a 	an instructor. to the portfolio course where "Add File" but required field at date). "Upload a File o upload a file second for a	o main page. the document will ton ls (type, subtype) a e" button. e format other than utomatic rendering	be imported and optional .pdf or .tex.	fields			
Expected result	The system message.	rejects unsup	ported file formats	and displays	s an error			
Date / Result	01/05/2025	- Pass						

Test ID:	FT037	Category:	Portfolio	Severity:	Major
Objective	Verify that th	ne system car	n create a new ass	ignment or p	ortfolio type
	(e.g., quiz o	r exam).			
Steps	1. Log in as	an instructor.			
	2. Navigate	to the portfoli	o main page.		
	3. Click the	course semes	ster where the assi	ignment type	will be created.
	4. Click the	"Add File" but	ton.		
	5. Click the	"Select Type"	bar and select "Ad	dd a new type	e".
	6. Type the	new assignme	ent type name in th	ne textbox that	at opens.
	7. Click the	"Add" button.			
Expected result	The system	successfully	creates the new as	ssignment typ	be with a
	success me	ssage and lis	ts the newly create	ed type in the	"Select Type"
	list in the "U	pload New Fi	le" modal.		
Date / Result	01/05/2025	- Pass			

Test ID:	FT038	Category:	Portfolio	Severity:	Moderate		
Objective	Verify that the system can successfully edit and update an existing syllabus in the database.						
Steps	 An instruct Navigate Click the Click the Click the Fill all the grade contri Click the 	ctor has alread to the main pa "Edit" button i "+" button to a required field bution) in the "Save" button	dy created a new o age of the selected n the course sumn add a new "assignr Is with valid data (n opened modal.	course. d course. nary. ment compor number of as	ient". signments,		
Expected result	The system syllabus in t saving.	successfully he summary s	updates the syllab section. A success	us and show message is	s the changed displayed after		
Date / Result	01/05/2025	- Pass					

Test ID:	FT039	Category:	Portfolio	Severity:	Moderate	
Objective	Ensure that an instructor can successfully create a merged portfolio of a course.					
Steps	 Log in as Navigate Click the Click the Select the Click the Click the Click the 	an instructor. to the portfoli course where "See Portfolic e desired docu "Create Portfo	o main page. the portfolio will b " button. uments from the lis blio" button. on in the open infor	e created. st. rmation moda	al.	
Expected result	The system success me	successfully ssage is disp	saves the merged ayed after saving.	document to	the system. A	
Date / Result	01/05/2025	- Pass				

Test ID:	FT040	Category:	Portfolio	Severity:	Minor			
Objective	Ensure that	Ensure that an instructor can successfully export a portfolio of a course.						
Steps	 Log in as Navigate Click the Click the Click the Select de Click the 	an instructor. to the portfoli course where "See Portfolic sired docume "Export" butto	o main page. the portfolio will b " button. nt(s) from the list. n.	e created.				
Expected result	The system local systen	successfully n. A success r	generates and exp nessage is display	oorts a portfol ed after expo	io file to the orting.			
Date / Result	01/05/2025	- Pass						

Test ID:	FT041	Category:	Portfolio	Severity:	Critical	
Objective	Verify that an instructor can delete a course, permanently removing all documents in the portfolio.					
Steps	 An instruct Log in as Navigate Click the Click the remove. Write "DE Click the 	ctor has alread an instructor. to the portfolio "Delete" butto ELETE" into th "delete" butto	dy created the cou o main page n from the action t e textbox in the po n, which will be clie	ab in the row p-up modal.	e. of the course	
Expected result	The system associated	permanently portfolio data.	deletes the course A success messa	e information ge is displaye	and all ed after deleting.	
Date / Result	01/05/2025	- Pass				

Test ID:	FT042	Category:	Portfolio	Severity:	Critical			
Objective	Verify that a	Verify that an instructor can delete a document permanently.						
Steps	 An instruct Log in as Navigate Click the Click the Click the Click the Click the 	ctor has alrea an instructor. to the portfoli course from v "See Portfolio "Delete" butto ow.	dy created the doc o main page. which the documen " button. n to remove from t	ument to ren t will be reme	nove. oved. o in the			
Expected result	The system message is	permanently displayed afte	deletes the docum er deleting.	ent informati	on. A success			
Date / Result	01/05/2025	- Pass						

Test ID:	FT043	Category:	Portfolio	Severity:	Moderate			
Objective	Verify that thas thas the second seco	Verify that the system can successfully edit and update an existing assignment in a syllabus in the database.						
Steps	 An instruct assignment Log in as Navigate Click the Fill all the the desired Click the 	 An instructor has already created a course with at least one type of assignment. Log in as an instructor. Navigate to the main page of the selected course. Click the "Edit" button in the course summary. Fill all the desired fields with valid data (date, no. of assignment) in the desired type of assignment. Click the "Save" button. 						
Expected result	The system successfully updates the assignment information and shows the changed information in the summary section. A success message is displayed after saving.							
Date / Result	01/05/2025	- Pass						

Test ID:	FT044	Category:	Portfolio	Severity:	Moderate				
Objective	Verify that th on the "Edit	Verify that the system can display an existing document in LaTeX format on the "Edit Exam" page.							
Steps	 There is a Log in as Navigate Click the Click the Click the Click the 	 There is at least one .tex type of assignment in the course portfolio. Log in as an instructor. Navigate to the portfolio main page. Click the course where the assignment will be opened. Click the "See Portfolio" button. Click the "Edit" button from the action tab in the document row to open. 							
Expected result	The system The LaTeX	The system successfully navigates the user to the "Edit Exam" section. The LaTeX content is correctly rendered and displayed.							
Date / Result	01/05/2025	- Obsolete							
Notes	This feature from this mo	This feature was already in the exam creation module, so it was moved from this module as a design choice.							

Test ID:	FT045	Category:	Portfolio	Severity:	Minor			
Objective	Verify that the system can successfully filter the portfolio according to the section.							
Steps	 An instruct Log in as Navigate Click the Choose t 	 An instructor has already created a course with at least one section. Log in as an instructor. Navigate to the portfolio main page. Click the "All Sections" button. Choose the desired section among the selections. 						
Expected result	The system successfully filters the portfolio page according to the section. A filtered page is displayed.							
Date / Result	01/05/2025	01/05/2025 - Pass						

5.2 Non-Functional Test Cases

Test ID:	NFT001	Category	Usability	Severity	Moderate					
Objective	Verify that the the the the the the term of term o	Verify that the UI of the software is self-explanatory so that users are not overwhelmed with excessive amounts of content at once.								
Steps	 Create sa Prepare a their usage Ensure th software. Evaluate 	 Create sample user groups. Prepare appropriate demo sessions for each group and observe their usage behaviors. Ensure they refer to the user manual and ask questions about the software. Evaluate the outputs. 								
Expected result	The user manual and UI should be adequate for users to perform the core actions of the software. A moderate number of questions is expected to be asked.									
Date / Result	01/05/2025	01/05/2025 / Pass								

Test ID:	NFT002	Category	Maintainability	Severity	Critical			
Objective	Verify that the code base is self-documented, non-redundant, and embracing new features.							
Steps	 Check for Check for Ask each developed. Determin 	 Check for redundant code segments. Check for code segments that do not conform to best practices. Ask each developer to cross-check the code blocks they have not developed. Ask them if those code segments are self-documenting. Determine risky parts that may block the development in the future. 						
Expected result	There are no risky code segments, or at least not a few.							
Date / Result	01/05/2025 / Pass							

Test ID:	NFT003	Category	Reliability	Severity	Critical				
Objective	Verify that th	Verify that the software is invulnerable to single-point failure.							
Steps	 Shut dow failures. Observe 	 Shut down services one at a time to identify possible single-point failures. Observe the outcomes on the client side. 							
Expected result	No or little in not cause a	No or little impact on the client side should be observed. Failures must not cause any sensitive data to leak.							
Date / Result	01/05/2025 / Pass								

Test ID:	NFT004	Category:	Security	Severity:	Critical				
Objective	Verify that c using Servio	Verify that only authorized services can communicate with each other using Service-to-Service Authentication (S2S Auth).							
Steps	 Identify a Service). From an send an aut Observe Repeat the observe the 	 Identify a service that requires S2S authentication (e.g., Objection Service). From an unauthorized service (e.g., Question Service), attempt to send an authenticated request to Objection Service. Observe the system's response. Repeat the request with a valid S2S authentication token and observe the response. 							
Expected result	The unauth Forbidden e	The unauthorized service receives a 401 Unauthorized or 403 Forbidden error with a message like "Service authentication failed."							
Date / Result	01/05/2025 / Pass								

Test ID:	NFT005	Category:	Security	Severity:	Critical
----------	--------	-----------	----------	-----------	----------

Objective	Verify that users can only access endpoints permitted by their role, ensuring Role-Based Access Control (RBAC) enforcement through JWT authentication.
Steps	 Log in as an Institution role user and obtain a valid JWT token. Make a POST request to the Instructor Register endpoint (/api/institution/register-instructors) using the Institution JWT token. Observe the response and confirm that the request is successful. Log in as an Instructor role user and obtain a valid JWT token. Attempt to make the same POST request to /api/institution/register-instructors using the Instructor JWT token.
Expected result	The Instructor role user receives a 403 Forbidden or 401 Unauthorized response with a message like "Permission denied."
Date / Result	01/05/2025 / Pass

6. Maintenance Plan and Details

Evalio's maintenance strategy focuses on proactive monitoring, incremental scaling, and controlled change management so that the platform stays reliable while new features are introduced.

6.1. Monitoring & Health Checks

Layer	Tool / Mechanism	What We Watch	Frequency
Micro-services	Consul health probes (`/health`)	200 OK & latency	10s
Container runtime	Docker events + log forwarding	Crash-loops, restarts	Real-time
EC2 nodes (prod)	AWS CloudWatch metrics	CPU, RAM, disk	-
GCS bucket	Object change notifications	Failed uploads	Real-time

Table 1: Monitoring & Health Check table of Evalio

6.2. Scheduled Maintenance Tasks

We plan to implement a structured maintenance routine that includes automated and manual tasks. These tasks aim to ensure system stability, security, and long-term performance as Evalio scales. Critical operations such as JWT key rotation and database backups are handled via cron jobs or cloud functions. Manual verification steps, like backup restoration, will also be part of the process.

Task	Frequency	Automated?	Owner
Rotate JWT signing keys	Monthly	Yes (cron job)	DevOps
Backup PostgreSQL	Daily @ 02:00 UTC	Yes	DBA
Verify backup restore on staging	Weekly	Manual	DBA
Re-index frequently accessed tables	Monthly	Yes (Alembic task)	DBA



6.3. Scaling & Capacity Planning

Since our architecture is already based on microservices, we plan to incorporate a container orchestration tool such as Kubernetes in the future to enhance scalability and reliability. Within this scope, we aim to implement the following improvements:

- Vertical Scaling: EC2 instances currently start at t3.medium. We plan to define Auto Scaling Group (ASG) policies that will scale up to t3.large when average CPU usage exceeds 70% for more than 10 minutes.
- Horizontal Scaling: All services are stateless and can be scaled out by increasing replica counts in docker-compose-infra.yaml. We plan to migrate this infrastructure to more dynamic and managed scaling as part of our roadmap.
- Auto Scaling: We intend to introduce auto-scaling policies based on real-time load metrics, ensuring that services can elastically respond to usage spikes without manual intervention [5].

7. Other Project Elements

This section explores the various elements of the Evalio.

7.1. Consideration of Various Factors in Engineering Design

7.1.1 Constraints

7.1.1.1 Authentication & Data Access

Evalio enforces 2FA to enhance security and prevent exam leaks. This additional layer of authentication is vital for protecting sensitive exam-related materials and maintaining the system's confidentiality. Evalio also implements RBAC to provide granular access to the system and data resources. Unauthorized entities can not access the protected resources. For instance, only instructors can access the question database, while teaching assistants can not view any questions.

7.1.1.2 Question Scanning

Evalio provides multiple ways of uploading questions to the system. Evalio allows instructors to upload questions using the LaTeX format. Questions are stored with various metadata in the database. In this way, questions can be edited more flexibly. Another way to upload a question is to scan it and pass it to the LLM. In that way, legacy questions can also be uploaded to the database.

7.1.1.3 Cloud Services & Containerization

Evalio is currently hosted on a cloud machine since establishing and maintaining a server from scratch is not viable for our team. Cloud services are the current trend, anyway. Evalio's services are fully dockerized, so it is easy to deploy and scale a complete application on a cloud machine, which is not a big deal.

7.1.1.4 Advertisement

Due to budget limitations and Evalio's target audience specificity, we advertise our application to a small instructor group. For the initial development phase of the application, this strategy did not overwhelm us with economic issues. Also, it allowed us to get significant feedback about the core functionality of the project.

7.1.1.5 Time Management

Since Evalio is a capstone project, its development and deployment calendar are limited by the academic calendar. We designed the application by considering the continuous deployment principle. Instead of implementing the project as a whole, we implemented it module by module. This way, we ensured that everything we had in production worked fully. At the end, we implemented the project incrementally.

7.2. Ethics and Professional Responsibilities

First, Evalio was designed as a project to become an industry standard. So we must provide a fair and inclusive environment to our users. So, in the first place, we designed Evalio's user interface in English to allow many users to use it. Even if Evalio is a for-profit application, it is not intended to undermine academic integrity. The primary purpose of the Evalio is to continually enhance the teaching and examination procedure for academic staff and students.

7.3. Teamwork Details

7.3.1. Contributing and functioning effectively on the team to establish goals, plan tasks, and meet objectives

As previously mentioned, our project was well-suited for a microservices architecture due to its modular structure. Therefore, we chose to partition the work as much as possible to maximize individual productivity, especially considering that we also managed coursework from other classes. As a result, frequent in-person collaboration was not always feasible.

To address this, we adopted the sprint methodology, meeting regularly to divide tasks, track progress, discuss challenges, and conduct joint testing. At the end of each sprint, we planned the functions for the upcoming sprint based on the project's progress and requirements.

Additionally, we regularly gathered feedback from our supervisor to ensure alignment with the project's objectives. Periodic requirement checks were also a priority to ensure all components remained on track and within scope.

7.3.2. Helping creating a collaborative and inclusive environment

We prioritized clear and professional communication to empower a supportive and collaborative environment. Tasks were regularly assigned through Jira during our weekly meetings. Once a task was assigned, the sprint leader consulted the designated team member to ensure they were comfortable with the scope and difficulty of the task, thereby avoiding potential overwork. Additionally, we ensured that if any member required assistance, other teammates were readily available to support them.

We utilized Zoom, WhatsApp, Gmail, and Jira to maintain an effective and seamless communication process. In cases of urgency, video calls enabled screen sharing and real-time problem-solving, greatly facilitating mutual assistance.

Furthermore, we recognized the importance of strong social bonds within the team. To strengthen these connections, we organized several in-person gatherings, such as team dinners and extracurricular activities, to build camaraderie beyond the project's scope.

7.3.3. Taking lead role and sharing leadership on the team

Our team adopted a dynamic leadership approach, with roles rotating seamlessly based on the project's needs. For instance, if the focus of a weekly meeting centered on a particular module, the team member responsible for implementing that module would assume leadership and guide the meeting accordingly.

One team member consistently shared their screen and managed task assignments through the Jira board, effectively taking on the project management responsibilities. We also held regular merge meetings to integrate all feature branches into the main branch. These sessions were conducted collaboratively to resolve any potential merge conflicts and to review the completed work at the end of each sprint.

During these merge meetings, one team member would take the lead by sharing their screen and managing the session from a single computer to maintain consistency and control throughout the process.

7.3.4. Meeting objectives

This section provides an overview of how each core functional module in Evalio has met the objectives set in the Analysis, and all essential modules have been implemented, integrated, and tested. Below is a module-by-module summary of functionalities and the level of completion.

Authentication & Authorization:

Planned Objectives

- User registration and login for different types of users.
- Role-based access control.
- Secure session management.

Achievements

- Full implementation of registration and login flows using email verification.
- JWT-based session management and secure access tokens are in place.
- Role separation is strictly enforced.
- All test cases related to authentication passed successfully.

Exam & Question Creation Module:

Planned Objectives

- Instructors can create and manage exam questions.
- Exams can be created by selecting questions.
- Versioning and editing of questions.

Achievements

- Instructors can create, edit, and delete questions.
- Exams can be assembled by selecting from existing questions.
- Drag-and-drop reordering of questions in exams is supported.
- Forking the versions of questions and redesigning are supported.
- All critical exam features are complete; minor UI arrangements may be needed (according to design choice).

Objection session scheduling & Slot Booking Module:

Planned Objectives

- TAs and Instructors can create objection sessions.
- TAs and Instructors define available time slots.
- Students can book slots.
- Prevent overlaps during slot booking.

Achievements

- Objection sessions can be scheduled per exam.
- Fully functional slot system.
- Booking requests are validated for conflicts and real-time access.
- Peer-to-peer connection is established only in booked sessions.

Courses & Portfolio Module:

Planned Objectives

- Instructors can create courses.
- Instructors can upload documents.
- Course materials can be uploaded and viewed securely.

Achievements

- Instructors can create and upload courses for specific semesters.
- Instructors can upload and manage documents.
- By using uploaded materials, the end-term report can be created.
- Unauthorized material access is blocked and tested successfully.
- Only TAs with permissions can access the Instructor's portfolio.

These developments align directly with the milestones and functional targets in the Analysis and Requirements Report.

In addition to functional requirements, the Evalio platform has successfully met all key ethics and professional responsibility objectives outlined in our initial project plan. Usability and accessibility design guarantee that the system maintains a clean, user-friendly interface and workflow and can be easily used by instructors and TAs. Instructor-specific portfolio views and institution-based course content are managed with strict control so TAs can only access course materials under the instructor's supervision. All modules from exam preparation to portfolio management and objection session coordination are hierarchically structured, with privacy and role-based data access importance. Sensitive academic data storage and handling are done with subject-controlled, secure mechanisms. Furthermore, Evalio includes all modules needed to assist instructors end-to-end in the exam preparation and documentation processes, including question creation, complete exam execution, objection session configuration, and preparing an end-term portfolio. These capabilities have been iteratively developed based on direct feedback from key users, so the platform closely aligns with their real-world needs.

7.4 New Knowledge Acquired and Applied

During the development of Evalio, we encountered several technologies and architectural patterns for the first time and had to learn and apply them effectively throughout the project. Although we had prior experience in web development, this was our first time building a microservice-based system with containerized services, centralized authentication, and distributed data management.

On the infrastructure side, we gained practical knowledge about Docker, Docker Compose, and how to build and deploy services in isolated containers. The use of Consul for service discovery and health checks was entirely new to us, and we learned how it helps ensure the resilience of a distributed system.

In addition, we became familiar with designing JWT-based role enforcement, handling token refresh logic on the frontend, and implementing RBAC (Role-Based Access Control) at both API and UI levels.

Beyond the technical scope, we also improved our project management and team collaboration skills. We followed an agile-inspired sprint structure and used Jira to plan, assign, and track tasks. This helped us manage a large codebase across multiple contributors and keep the team aligned throughout the development lifecycle.

Most of this knowledge was gained through hands-on debugging, following tutorials, reading official documentation, consulting with our team and consulting with our advisor. Overall, this project has strengthened both our technical and collaborative understanding of building and managing modern cloud-native applications.

8. Conclusion and Future Work

8.1 Conclusion

Evalio is structured as a safe, modular, microservices-based system. Its core functionality of question building, LaTeX rendering for exams, booking objection sessions, and managing portfolios is written as independent, scalable services. The system uses Docker, access control through JWT, Redis caches, and cloud storage to achieve performance, safety, and supportability. The architecture is future-proofed after all essential modules have been tested correctly.

On the user requirements side, Evalio prioritizes usability, privacy, and academic integrity. Role-based UI allows instructors, TAs, and institutions to operate in a well-defined scope. Real academic workflows, such as building question banks, managing objections, and report generation, are all fully supported. The system reflects user feedback and includes helpful tools augmenting teaching, testing, and collaboration.

8.2 Future Work

Although Evalio successfully passed most security-related non-functional requirements, it failed specific attack vector tests (e.g., XSS, SQL Injection). Therefore, a critical direction for future work is to strengthen security further. This includes sanitizing input completely, applying HTTP security headers, and regularly conducting audits so that the system can stay resilient against new attacks.

From a functional standpoint, the platform can be enhanced by enabling instructor-TA interactions on assignments. For example, when a teaching assistant uploads a quiz or homework assignment, the instructor should be able to provide

direct feedback, request revisions, or post contextual comments on the file. This feature would encourage collaborative content refinement and quality assurance.

As one part of making student access continue without needing another server or platform, instructors could optionally share certain documents (e.g., lecture notes or assignment files) with students via secure, token-based links, similar to the objection module's access flow. This would provide controlled yet direct access to educational material.

Lastly, a notification system can be integrated to inform users of relevant actions (e.g., objection booking updates, portfolio submissions, or shared document access), which provide overall responsiveness and communication within the system.

9. User Manual

9.1 General

9.1.1 Landing Page



Figure 24: Landing Page

The landing page of Evalio provides an informative and straightforward interface for users. It shows the main capabilities of the application, how users can contact us, and the possible subscription options. Users can go to the developers page, login page, and registration request page from the navigation bar by clicking the respective buttons.

🧭 evalio		Team	Login	Registration Request	Purchase	Contact
	Cogin					
* Email A Er	er your email					
* Pasa @ E	ord er your password					
•	Login					
	Forgot Password?					

9.1.2 Login

Figure 25: Login Page

The login page asks for the user's username and password. After entering the correct credentials, the user is authorized. Otherwise, the page shows a warning message.

9.1.3 Registration Request

🌶 evalio			Team Login	Registration Request	Purchase	Con
Particularization ● Instructional ● Instructio	Free Trial Free for 15 Days Free for 15 Days Valid for 15 days July to 5 instructors Di exam creations Basic support	 Sitver Subscription S900/year Register up to 20 instructors Register up to 10 teaching assistants Control Control Control	Gold S S1 All Silver Register 5 Mutti-insti Priority su	Subscription 5000/year exturus included 0+ instructors 0+ teaching assistants tudion question sharing opport		

Figure 26: Registration Request Page

evalio				Team Login Registration Request	Purchase Co
	Vour request has been sent successfully: No will send your request to one of our administ. One approved, you will be directed administ. One approved you will be directed administration of the administration of the board.	Free Trial Free for 15 Days Free for you to know us better Valid for 15 days Vulo for 15 days 10 ps instructors 10 exam creations Basic support	Sitver Subscription S000/year P Register up to 20 instructors P Register up to 20 instructors P Register up to 21 instructors Unlimited question database Unlimited question database P Full LaTeX/PDF support	Back Subscription S15000year All Silver fautures included All Silver f30+ instructors All Silver f30+ instructors Multi-institution question sharing Multi-institution question sharing Priority support	

Figure 27: Registration Request Response

In the registration request page, institutions can send a registration request to Evalio by filling in the necessary information and clicking the "Send Your Request" button. When a request is submitted, the page shows an information message.

9.2 Admin Management

9.3.1 Admin Page

🧭 evalio				Manage G
Admin Management Manage your institution's settings and users here. List institutions Institution Requests Add Tag				
а	Name	Email	Last Login	
6aa8750a-944a-4b55-ab11-54616cfa24cd	Default Institution	ins@example.com		
d7927a78-487c-4ce5-9d3a-1c6cb9375624	Bilkent	yusuf.toraman@ug.bilkent.edu.tr	4/29/2025, 10:24:10 PM	
				< 1 >

Figure 28: Admin Management Page, List Instructor Tab

Evalio admins can see the list of existing institutions with their IDs, names, emails, and last login dates. All institutions can be viewed in a paginated table. Pagination is controlled from the arrows at the bottom of the table.

🥑 evalio				Manage 🤤
Admin Management				
Manage your institution's settings and users here.				
List Institutions Institution Requests Add Tag				
Name	Email	Status	Actions	
Karabük Üniversitesi	k78@example.com		Approve Reject	
				< 1 >

Figure 29: Admin Management Page, Add Instructor Tab

Evalio admins can see the list of registration requests. The institution is officially registered if the admin clicks the "Approve" button. The request is rejected if the admin clicks the "Reject" button. All requests can be viewed in a paginated table. Pagination is controlled from the arrows at the bottom of the table.

🧭 evalio	Manage	G
Admin Management		
Manage your institution's settings and users here.		
List Institution Requests Add Tag		
Add Tag Tag Name + Add Tag		
Existing Tags		
Tag Name		¢
Binary Search		
HashTable		
AVL Tree		
Greedy Algorithms		
Dynamic Programming		
	< 1	2 >

Figure 30: Admin Management Page, Add Tag Tab

Evalio admins can add tags for questions. Admin enters the category's name and hits the "Add Tag" button. A new tag is listed in the Existing Tags table if a tag can be added. Otherwise, an error message is shown. Admins can also list the existing tags in the Existing Tags table in a paginated fashion.

9.3 Profile Pages

9.3.1 Institution Profile Page

) evalio						Profile
	Instructors List Teaching As	sistans List Course List				
Institution Profile Name: Default Institution	Name	Status	Email	Last Login	Actions	
Email: ins@example.com Total Exams: 1	Default Instructor	ACTIVE	int@example.com	-	Delete Ban	
Total Questions: 2						< 1 >
Total TAs: 1						
Actions						
Add Members						
Add Course						

Figure 31: Institution Profile Page, Instruction List Tab

evalio						Profile
	Instructors List Teaching Assistans List	Course List				
Institution Profile	Name	Status	Email	Last Login	Actions	
Email: ins@example.com Total Exams: 1	Default Teaching Assistant	ACTIVE	ta@example.com	-	Delete	Ban
Total Questions: 2						
Total Instructors: 1 Total TAs: 1						
Actions						
Add Members						
Add Course						

Figure 32: Institution Profile Page, Teaching Assistant List Tab

) evalio					Profile
	Instructors List Teaching Assistans List Course List				
Institution Profile Name: Default Institution	Course Name	Course Code	Description	Department	Actions
Email: ins@example.com	Automata Theory And Formal Language	CS476	Formal languages	CS	0
Total Exams: 1 Total Questions: 2					
Total Instructors: 1					
Total TAs: 1					
Actions					
Add Members					
Add Course					

Figure 33: Institution Profile Page, Course List Tab

Institutions can manage their profile in the Institution profile page. On the left side, several stats about institutions are listed. Under the information, institutions can add new members by clicking the "Add Members" button and add new courses by clicking the "Add Course" button. Institutions can list, block, and delete their

instructors and teaching assistants from "Instruction List" and "Teaching Assistant List", respectively. Institutions can list their courses under the "Course List" tab.

evalio					Profile
		In: Add Member			
nstitution Profile		Single User Upload CSV	Dercriptio	Department	Actions
lame: Default Institution				bepartment	Petions
mail: ins@example.com		Enter name			
otal Exams: 1		Enter email			
otal Questions: 2		Instructor	×		
ntal TAs: 1		Add User			
ations.					
	Add Members				

Figure 34: Institution Profile Page, Add Single Use Modal

🧭 evalio		Profile G
Institution Profile	In Add Member Single User Upload CSV	Antina
Name: Default Institution Email: Ins@example.com Total Exemt 1 Total Question: 2 Total Instructors: 1 Total Instructors: 1	instructor v L Upload CSV r () () () () () () () () () (ALIMU
Actions Add Members Add Course		
Add Course		

Figure 35: Institution Profile Page, Upload CSV Modal

After clicking the "Add Members" button, a modal pops up. Institutions can add either a single user via the "Single User" tab or add bulk via "Upload CSV". The Institution fills in the necessary information for single-user addition and clicks the "Add User" button. The Institution prepares a CSV file containing information on multiple users for bulk ads. Then uploads it to the system.

🧭 evalio				Profile G
	In Add Course			
Institution Profile	Enter course name		1	Lan
Name: Default Institution	Enter course code	Description	Department	Actions
Email: ins@example.com	Enter course description			
Total Exams: 1				
Total Questions: 2	Enter department			
Total Instructors: 1	Add Course			
Total TAs: 1				
Actions				
Add Members				
Add Course				
Add course				

Figure 36: Institution Profile Page, Add Course Modal

After clicking the "Add Course" button, a modal pops up. The institution fills in the necessary course information. Then click the "Add Course" button.

9.3.2 Instructor Profile Page

Instructor Profile Name: Default Instructor Email: int@example.com Institution: Default Instructor Institution: Default Instructor Institution: Default Instructor Institution: Default Instructor Actions	Actions D C
Instructor Profile Class Name Status Nume: Default Instructor Midterm Objection Title Class Name Status Institution Institution Finaliting/example.com CS426 TERMINATED Actions Actions Actions Actions	Actions D C
Email: int@example.com Midterm Objection C5426 TERMINATED Institution Institution Email: in@example.com Rele: instructor Adderm Objection C5426 TERMINATED	0
Institution Institution Default Institution Institution Email: Ind@example.com Role: Instructor	< 1
Actions	
Create Objection Session	
Create New Exam	
Create New Question	

Figure 37: Instructor Profile Page, Objection List Tab

🧭 evalio					Questions	Courses and Portfolio	Profile G
	Created Objections	Created Exams Created Questions	Forked Questions	Favorited Questions			
Instructor Profile Name: Default Instructor Email: Int@example.com	Exam Title	1	Status		Date		
	CS476 - CW3		draft		01/05/25 18:55:23		
Institution Institution: Default Institution Institution: Email: Ins@example.com Role: Instructor							< 1 >
Actions							
Create Objection Session							
Create New Exam							
Create New Question							

Figure 38: Instructor Profile Page, Exam List Tab
evalio					Questions	Courses and Portfolio	Profile
	Created Objection	ons Created Exams	Created Questions Forked Question	ons Favorited Questions			
Instructor Profile Name: Default Instructor	Title	Difficulty	¢ ▼ Tags	Actions			
Email: int@example.com	test	Medium	Array	Add to Exam	Y Fork	Stats	
Institution Institution: Default Institution Institution: Email: ins@example.com Role: Instructor							< 1
Actions							
Create Objection Session							
Create New Exam							
Create New Question							

Figure 39: Instructor Profile Page, Questions List Tab

Instructor Profile							Questions	Courses and Portfolio	Profile
Instructor Profile		Created Objections	Created Exams	Created Questions	Forked Questions	Favorited Questions			
Name: Default Instructor		Title	Difficulty		¥ Tags		Actions		
imail: int@example.com		test	Medium		Аггау		h Show	Parent	
nstitution nstitution: Default Institution nstitution Email: ins@example.com Role: Instructor									< 1
Actions	an Garrier								
Create Object	w Exam								
Create New	Question								

Figure 40: Instructor Profile Page, Forked Questions List Tab

evalio					Questions	Courses and Portfolio	Profile
	Created Obje	ctions Created Exams	Created Questions Forked Que	estions Favorited Question	ns		
Instructor Profile Name: Default Instructor	Title	Difficulty	≎ ¥ Tags	Actions			
Email: int@example.com	test	Medium	Аггау	Add to Exam	Y Fork Let St	ats	
Institution Institution: Default Institution Institution Email: ins@example.com Role: Instructor							< 1
Actions							
Create Objection Session							
Create New Exam							
Create New Question							

Figure 41: Instructor Profile Page, Favorites Question List Tab

In the Instructor Profile page, instructors can see information about themselves. On the left side of the page, the Instructor can see information about them and their institution. Under the "Actions" card, they can create objections, exams, and questions by clicking the "Create Objection Session", "Create New Exam", and "Create New Question" buttons, respectively. On the right side of the page, instructors can see the list of objections, exams, questions, forked questions, and favorited questions from "Created Objections", "Created Exams", "Created Questions", "Forked Questions", and "Favorited Questions" tabs, respectively.

9.3.3 Teaching Assistant Profile Page

🗸 evalio		Create Objection	Courses and Portfolio	Profile G
	Created Slots Assigned Exams			
aching Assistant Profile er Default Teaching Assistant It sa@example.com	Objection Title	Class Name	Status	
	CS476 - Automata Theory and Formal Languages	EE05-03	TERMINATED	
Institution	CS315 - Programming Languages	EAZ03	ACTIVE	
Institution: Default Institution Institution Email: ins@example.com Role: Teaching Assistant				< 1 >
Actions				
Create Objection Session				

Figure 42: Teaching Assistant Profile Page, Objection List Tab

🧭 evalio			Create Objection Courses and Port	folio Profile G
	Created Slots Assigned Exams Utabiling Assistant tample.com Default Institution finals insel@example.com ing Assistant Created Slots Assigned Exams Tate CST76 - CW3			
Teaching Assistant Profile Name: Default Teaching Assistant	Title	Status	Date	
Email: ta@example.com	CS476 - CW3	draft	01/05/25 18:55:23	
Institution Institution: Default Institution Institution Email: Ins@example.com Role: Teaching Assistant				< 1>
Actions Create Objection Session				

Figure 43: Teaching Assistant Profile Page, Assigned Exams List Tab

In the Teaching Assistant Profile page, TAs can see information about themselves. On the left side of the page, the TA can see information about them and their institution. Under the "Actions" card, they can create objections via the "Create Objection Sessions" button. On the right side of the page, teachers can see the list of objections and assigned exams from the "Created Slots" and "Assigned Exams" tabs, respectively.

9.4 Question & Exam Module

🥑 evalio						Questions	Courses and Portfolio	Profile	G
Question L	ist	Reset							
Title 🗘	Difficulty 🗘 🛡	Tags 🛛 🔻	Average Success %	Institution T					
test	Easy	Binary Search	-	Default Institution (Forked)	Add to Exam	Y Fork	lat Statistics		
test	Easy	Binary Search		Default Institution (Forked)	Add to Exam	¥ Fork	lat Statistics		
test	Easy	Binary Search		Default Institution (Forked)	Add to Exam	Y Fork	Int Statistics		
test	Easy	Binary Search		Default Institution	Add to Exam	Y Fork	lat Statistics		
test	Medium			Default Institution	Add to Exam	Y Fork	lat Statistics		
integraș	Easy	Binary Search		Default Institution (Forked)	Add to Exam	Y Fork	lat Statistics		
									•••

9.4.1 List Questions

Figure 44: Question List Page

The Question List page displays all available questions in a table format. Users can search for questions using the search bar and reset filters with the "Reset" button. Each question is listed with its title, difficulty level, tags, average success rate, and institution. On the right side, users can take actions such as adding the question to an exam, viewing its versions, forking it for edits, or checking statistics. Difficulty levels are shown with colored labels. Pagination controls at the bottom let users navigate through multiple pages of questions.

9.4.2 Create Question

🧭 evalio	Questions Courses and Portfolio Profile
Add Question	
* Title	
Enter question title	
* Difficulty	
* Required Space (Lines)	
Tags	
Add tag v	No LaTeX file available for this question.
Description	
Enter question description	
Is Private0/500 characters	
🗅 Export LaTeX File 🛛 🕒 Import LaTeX File 🕞 Convert Image to LaTeX 🖉 🖉 Edit on LaTeX Editor	
Add Question Reset × Cancel	

Figure 45: Create Question Page

The Create Question page allows users to add new questions to the system. On the left, users fill in the question's title, select its difficulty level, specify the required space in lines, assign relevant tags, and optionally write a description. There's also a checkbox to mark the question as private.

At the bottom, users can export or import LaTeX files, convert an image to LaTeX, or open a built-in LaTeX editor to write the question content. Once all fields are completed, clicking Add Question saves it to the database. The Reset button clears the form, and Cancel exits the page without saving. If available, a preview area displays the LaTeX content converted to PDF on the right side.

9.4.3 Show Question Details



Figure 46: Show Question Details Page

The Show Question Details page displays all information about a specific question. On the left, users can see the question's title, difficulty level, visibility status, required space in lines, average success rate, description, and associated tags. Below, several action buttons allow users to edit the question, view its parent version, fork it, check statistics, add it to an exam, or export it as a PDF or LaTeX file. On the right side, the LaTeX-rendered version of the question is shown in a PDF viewer, allowing users to preview exactly how it will appear in exams or documents.

9.4.4 Edit Question

🔮 evalio	Questions C	ourses and Portfolio Profile C
Update Question	⊖ 130% ₹ ⊕	\$ + £ & ⊜ i
* Tile (integral		
Difficulty Easy V	1.Is it true that $x^n + y^n = z^n$ if x, y, z and n are positiv 2. Prove that the real next of all non-trivial zeros of the	e integers? Explain. function / (a) is 1
Required Space (Lines) 100 Tans	3.Compute $\int_0^\infty \frac{\sin(x)}{x} dx$	murion (s) n 2
Brauy Search Add tag v		
Description Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis aliquet ultrices viverra. Fusce id mauris nunc. Phasellus sit amet tortor aliquet, aliquet lacus quis, lacinia urna. Lorem ipsum dolor sLorem ipsum dolor st amet, consectetur adipiscing elit. Duis aliquet ultrices viverra. Fusce id mauris nunc. Phasellus sit amet tortor aliquet, aliquet lacus quis, lacinia urna. Lorem ipsum dolor 123.orem ipsum dolor sit amet, consectetur adipiscing elit. Duis aliquet ultrices mauris		
500500 characters Is Private Useport LaTeX File Convert Image to LaTeX Edit on LaTeX Editor Convert Image to LaTeX Edit on LaTeX Editor Convert		

Figure 47: Update Question Page - Question Data



Figure 48: Update Question Page - LaTeX Editor

The Edit Question page lets users modify all details of an existing question. Users can update the title, difficulty, required space, tags, and description. If the description exceeds the 500-character limit, an error is shown. On the right, the LaTeX preview shows how the question will appear in formatted form. By clicking Edit on the LaTeX Editor, users can directly modify the LaTeX code in a built-in editor. This allows for

precise formatting and math notation. Any changes made are instantly reflected in the live preview. Once edits are complete, users can click Update Question to save changes.

🧭 evalio		Questions	Courses and Portfolio	Profile	G
	Create an Exam				
	Exam inte:				
	Enter exam tue				
	Duration (minutes):				
	30				
	Exam Date:				
	Course:				
	Select a course V				
	Description:				
	Enter a brief description				
	005175				
	CREATE				

9.4.5 Create New Exam

Figure 49: Create New Exam

The Create Exam page is used to set up a new exam. Users must enter the exam title, minute duration, exam date, the related course, and a brief description. All fields are required for the exam to be created properly. Once the form is filled out, clicking the "Create" button finalizes the setup and registers the exam in the system.

9.4.6 Add Question To Selected Exam

🧭 evalio	,						Questions	Courses and Portfolio	Profile G
Question L Search questions	List globally	Reset		Add "integral"	to Exam				
Title \$	Difficulty 🗘 👻	Tags ¥	Average Success 9	test					
integral	Easy	Binary Search			Default Institution (Forked)	Add to Exam	Versions Y Fork	Le Statistics	
test	Easy	Binary Search			Default Institution (Forked)	Add to Exam	Versions Y Fork	Le Statistics	
test	Easy	Binary Search			Default Institution (Forked)	Add to Exam	Versions Y Fork	Les Statistics	
test	Easy	Binary Search			Default Institution (Forked)	Add to Exam	Versions Y Fork	Le Statistics	
test	Easy	Binary Search			Default Institution	Add to Exam	Versions Y Fork	Les Statistics	
test	Medium				Default Institution	Add to Exam	Versions Y Fork	Let Statistics	
									< 1 >

Figure 50: Add Question to Exam Pop-Up

When you click "Add to Exam" for a question in the Question List, a pop-up allows you to assign that question to an existing exam. From the dropdown menu, select the target exam by its name. Once selected, the question will be added to that exam's question pool. This feature helps you quickly build exams by selecting and inserting questions from the central repository.

9.4.7 Show Exam Details

🧭 evalio		Questions Courses and Portfolio Profile
Exam Details	Q ^ 1 /2 V	Q 130% V Q D + 2 & §
Exam Name Lest		test
Course cs315 - newtork Duration (minutes) O minutes		Question Number Assigned Grade Grade i 0.0
Status draft		
Description 123 Exam Instructions		
Questions Q. Search an	estion to Exam	
+ Question Number Title Tags Assigned Grade A	ccess %	
1 test (Binary Search) 0 0		
🖉 Edit Exam 📑 Delete Exam 🔄 Export Exam LaTeX 🔄 Export Exam PDF		

Figure 51: Exam Details Page

The Exam Details page shows all the information related to a selected exam. On the left, it displays the exam name, course, duration, status (e.g., draft), description, and any instructions provided. Below, it lists all the questions added to the exam along with their number, title, tags, assigned grade, and average success rate. Users can search for and add more questions using the button above the list. At the bottom are options to edit or delete the exam and export it as a LaTeX file or a formatted PDF. On the right, a live PDF preview shows how the exam will appear when printed or shared.

9.4.8 Edit Exam Details

🍼 evalio									Questions	Cou	urses and Portfoli	o Profi	le 🥃
Exam Details						4	1 /3	및 15	196 ¥ 🖽		q	* + 포	2 = :
Exam Name						8							
test						Ø							
Course									test				
cs315 - newtork					×								
Duration (minutes)								Question Number	Assigned Grade	Grade			
30								1	0.0				
Status								2	0.0				
Draft					~			· ·	5.0				
Description													
123													
Exam Instructions					li li								
					6								
Questions				Q Search and A	Add Question to Exam								
+ Question Number	Title	Tags	Assigned Grade	Average Success %	Action								
1	test	Binary Search	0	0	Remove								
2	test	Binary Search	U	U	Remove								
													-
Save Changes X Ca	ancel												

Figure 52: Edit Exam Details Page

The Edit Exam Details page lets users modify the content and structure of an existing exam. Users can update the exam name, related course, duration, status, description, and instructions. The list of added questions is shown below, where users can adjust assigned grades or remove questions entirely.

The order of questions can be rearranged easily using drag-and-drop functionality, allowing instructors to organize the exam flow as desired. Additionally, selecting a question in the list opens it for inline editing. This editing affects only the version of the question within the exam—it does not alter the original question in the central question bank. After making changes, users can click Save Changes to apply them or Cancel to discard edits. On the right side, a live preview shows the updated PDF version of the exam.

9.5 Objection Module

9.5.1 Creating an Objection Session (TA/Instructor)

🔮 evalio		Create Objection	Courses and Po	ortfolio Profile G
	Created Slots Assigned Exams			
Teaching Assistant Profile Name: Default Teaching Assistant	Objection Title	1	Class Name	Status
Email: ta@example.com	CS476 - Automata Theory and Formal Languages		EE05-03	TERMINATED
Institution	CS315 - Programming Languages		EAZ03	ACTIVE
Institution: Default Institution Institution Email: ins@example.com Role: Teaching Assistant				< 1 >
Actions				
Create Objection Session				

Figure 53: Profile Page of TA/Instructor with the "Created Slots" tab active

Log in as a TA or Instructor. On the Profile Page, locate the Actions section in the information card on the left side. Click the "Create Objection Session" button. You will be redirected to the session creation page upon clicking the button.

evalio		Create Objection	Courses and Portfolio	Profile	
	Create an Objection Session	n			
	Location:				
	Enter class name				
	Objection Title:				
	Enter objection title				
	Slot Duration (minutes):				
	30				
	Slot Capacity:				
	1				
	Day:				
	O Add Another Day				
	CREATE				

Figure 54: Create Objection Session Page

Complete the form on the "Create an Objection Session" page by providing all the required information. The Slot Capacity field specifies the maximum number of students who can reserve a single time slot. You may select either a single date or multiple dates for scheduling objection sessions. To add a date, click the "Add Another Day" button.

The Time Picker allows you to define the session duration precisely. For instance, if each slot is 30 minutes long, setting the start and end times at either xx:00 or xx:30 is recommended to avoid scheduling errors.

9.5.2 Managing the Objection Session (TA/Instructor)

0		Create Objection	Courses and Portfolio	Profil
CS476 - Automata	Theory and Formal Languages	Objection Session		
Reveal Session Link	minate Session			
As a teacher/TA/institution, you c	an disable or enable slots, or see which students hav	re reserved them.		
Classroom(s): EE05-03 TA/Instructor: ta@example.com Each slot can have: 2 student(s)				
Session status. ACTIVE	← 01/05/2025	(Day 1 of 1) \rightarrow		
	Time	Slot Info		
	11:00	AVAILABLE		
	11:15	AVAILABLE		
	11:30	AVAILABLE		
	11:45	AVAILABLE		
	12:00	AVAILABLE		

Figure 55: Objection Session Page

After clicking the "CREATE" button, you will be redirected to the Objection Session page, where you can copy the session link, enable or disable slots, and view reservations. To copy the session link, click the "Reveal Session Link" button and copy the displayed link. You can then share it with your students via email or another preferred communication method.

If the session includes multiple days, you can navigate between them using the arrow buttons at the top of the table to change the displayed date. The page also displays session details such as the assigned classroom(s), the creator of the session, slot capacity, and current session status.

evalio		Create Objection	Courses and Portfolio	Profile
CS476 - Autom	ata Theory and Formal Languages	Objection Session		
Reveal Session Link	Terminate Session			
As a teacher/TA/institution,	you can disable or enable slots, or see which students hav	e reserved them.		
Classroom(s): EE05-03				_
TA/Instructor: ta@example	.com			
Each slot can have: 2 stud	ent(s)			
Session Status: ACTIVE				
	- 01/05/2025	(Day 1 of 1) \rightarrow		
	Time	Slot Info		
	11:00	AVAILABLE		
	11:15	Click to disable		
	11:30	AVAILABLE		
	11:45	DISABLED		
	12:00	AVAILABLE		
	12:15	AVAILABLE		
	12:30	AVAII ABI F		

Figure 56: Objection Session Page

By clicking the "Terminate Session" button, the session will be simultaneously terminated for all users. If you click on an available slot cell, you can disable it. Similarly, if you click on a disabled cell, you can enable it.

		Create Objection	Courses and Portfolio	Prof
00470	- The same and Farmer 11 are seen	- Ohiostion Coopien		
CS476 - Automata	mineto Socion	s Objection Session		
As a teacher/TA/institution, you	can disable or enable slots, or see which students h	ave reserved them.		
Classroom(s): EE05-03				
Fach slot can have: 2 student	ר בו			
Session Status: ACTIVE	*/			
	← 01/05/20 Time	25 (Day 1 of 1) →		
	11:00	AVAILABLE		
	11:15	Click to see reservations		
	11:30	AVAILABLE		
	11:45	DISABLED		
	11:45	FULL		

Figure 57: Objection Session Page

Session Status: ACTIVE			
	Slot Reservations		
	reset.demir@ug.bilkent.edu.tr		
	hayrettin.arim@ug.bilkent.edu.tr		
	11:00	AVAILABLE	
	11:15	FULL	
	11:30	AVAILABLE	
	11:45	DISABLED	
	12:00	FULL	
	12:15	AVAILABLE	
	12:30	AVAILABLE	
	12:45	AVAILABLE	
	13:00	AVAILABLE	
	13:15	AVAILABLE	

Figure 58: Objection Session Page with reservations pop-up

If you click on a cell containing at least one reservation, you can view the corresponding reservations for that particular time slot. This feature is handy during the physical session, allowing you to verify which students have made reservations.

)		Create Objection	Courses and Portfolio	Prof
CS476 - Automata	a Theory and Formal Language	s Objection Session		
Classroom(s): EE05-03 TA/Instructor: ta@example.co Each slot can have: 2 student(Session Status: TERMINATED	n s)			
	← 01/05/202	25 (Day 1 of 1) 🛛 🔿		
	Time	Slot Info		
	11:00	0 students reserved		
	11:15	2 students reserved		
	11:30	0 students reserved		
	11:45	0 students reserved		
	12:00	2 students reserved		
	12:15	0 students reserved		

Figure 59: Terminated Objection Session Page

Once you click the "Terminate Session" button, the table will change color to indicate that the session has been closed, and no further modifications can be made. The page becomes read-only for instructors and TAs; however, they can still view existing reservations by clicking on the slot cells.

S evalio		Team	Login	Registration Request	Purchase	Contact
	Enter Your Email					
	Please provide your email to continu	ie.				
	Your university email					
	Continue					
	_					

9.5.3 Making Objection Session Reservations (Student)

Figure 60: Enter-Email Page (student view)

Once your instructor or TA shares the objection session link with you, you can access the system and reserve a slot for your objection session. Enter your email address and click the "Continue" button to proceed.

CS315 - Programm Select an available slot to reserve	ing Languages Objection Ser	ssion		
Classroom(s): EAZ03 TA/Instructor: ta@example.com Each slot can have: 2 student(s) Session Status: ACTIVE				
	← 01/05/20	D25 (Day 1 of 1) \rightarrow		
	Time	Slot Info		
	10:00	FULL		
	10:30	2 AVAILAB	LE	
	11:00	1 AVAILAB	.E	
	11:30	2 AVAILAB	E	

Figure 61: Objection Session Page (student view)

After entering your email, you will be directed to the Objection Session page. On this page, you can view disabled slot cells, available slots, their remaining capacity, and fully booked slots. To make a reservation, simply click on any available slot cell.

CS315 - Programm	ing Languages Objection S	Session	
Select an available slot to reserv	e your session.		
Classroom(s): EAZ03 TA/Instructor: ta@example.com Each slot can have: 2 student(s; Session Status: ACTIVE			
	← 01/05	5/2025 (Day 1 of 1) \rightarrow	
	Time	Slot Info	
	10:00	FULL	
	10:30	2 AVAILABLE	
	11:00	RESERVED	

Figure 62: Objection Session Page (student view)

Once you make a reservation, your selected slot cell will turn green, indicating a successful booking. You cannot select another slot unless you cancel your current reservation. To do so, simply click on your reserved cell and choose the option to cancel your reservation.

Cace Reservation Set can available slot to reserve your session Cace Reservation No Cace Reservation	evalio		Team	Login	Registration Request	Purchase	Contact
Select an available slot to reserve your session No Cancel Reservation Elassroom(s): EA203 EAch slot can have: 2 student(s) Each slot can have: 2 student(s) Session Status: ACTIVE \checkmark 01/05/2025 (Day 1 of 1) \rightarrow \rightarrow Image: tank of the tank of	CS315 - Program	Cancel Reservation Are you sure you want to cance	el your reservation?				
Classroom(s): EA203Education Laboration L	Select an available slot to re	serve your session	No Cancel Reserv	vation			
 ← 01/05/2025 (Day 1 of 1) → Time Slot Info 10:00 FULL 10:30 2 AVAILABLE 11:00 RESERVED 11:30 2 AVAILABLE 	Classroom(s): EAZ03 TA/Instructor: ta@example Each slot can have: 2 stude Session Status: ACTIVE	com nt(s)					
Time Slot Info 10:00 FULL 10:30 2 AVAILABLE 11:00 RESERVED 11:30 2 AVAILABLE		← 01/05/	2025 (Day 1 of 1) 🛛 🔿				
10:00 FULL 10:30 2 AVAILABLE 11:00 RESERVED 11:30 2 AVAILABLE		Time	Slot	: Info			
10:30 2 AVAILABLE 11:00 RESERVED 11:30 2 AVAILABLE		10:00	FL	JLL			
11.00 RESERVED 11.30 2 AVAILABLE		10:30	2 AVA	ILABLE			
11:30 2 AVAILABLE		11:00	RESE	RVED			
		11:30	2 AVA	ILABLE			

Figure 63: Objection Session Page (student view) with Cancel Reservation Pop-up

You can reserve any other available slot once you click the "Cancel Reservation" button. Please note that you can have at most one active reservation at anytime.

9.6 Portfolio Module

Current Courses						Add New Cours	e Filter
Course Name	•	Course Code	Department	Samester	Actions		
Network		CS421	cs	2026 - Spring	Detail	Edit D	
							1>
All Courses						Add New Cours	e Filter
All Courses	\$	Course Cade	Department	Semaster	Actions	Add New Cours	e Filter
Course Name Network	8	Course Code CS421	Department	Semaster 2025 - Spring	Actions Detail	Add New Cours	e Filter

9.6.1 Courses and Portfolio Page

Figure 64: Image of Courses and Portfolio Page

The Course and Portfolio page is the Portfolio Service's entry point and can be accessed via the "Courses and Portfolio" item in the top navigation bar. The user ("Instructor or TA") can view their previously added courses on this page.

The top section ("Current Courses") displays the courses the user is actively teaching this semester. The bottom section ("All Courses") lists all courses the user is related to, including past and future semesters. For example, CS421-2025 Spring appears in both sections since it is currently active, while CS421-2023 Fall only appears in the "All Courses" section, indicating that it is not currently active.

Users can perform several actions from this page. Clicking the Add New Course button opens the form to register a new course, while the Filter button allows courses to be filtered by semester and year. The most crucial action is entering the course's detail page using the Detail button, which redirects to the Course Summary Page. Additionally, the Edit button allows updating course metadata, and the Delete icon removes the course from the user's list.

Each of these interactions is described in detail in the following figures.

evalio						Que	estions	Courses and Portfolio	Profile	
Portfolio			Add New * Course	Course						
Current Courses			Select a d	course		v		Add New Cou	rse Filter	
Course Name	÷	Course Code	* Year 2025			8	Actions			
Network		CS421	* Semester				Detail	Edit D		
			Select se	mester		V			< 1 >	
			* Number o	f Sections						
			2							
All Courses			* Set Cours Yes	e Status 🕕		×		Add New Cou	rse Filter	
Course Name	¢				Cancel	Add	Actions			
Network		CS421		CS	2025 - Spring		Detail	Edit D		

9.6.2 Add New Course Modal

Figure 65: Image of Add New Course Modal

When users click the Add New Course button on the Courses and Portfolio page, this modal appears. It allows the useThis modal appears when users click the Add New Course button on the Courses and Portfolio pagesemester, and number of sections. The "Set Course Status" field determines whether the course is currently active (Yes) or not (No). Once all fields are filled, clicking the Add button registers the course under the user's portfolio, while Cancel closes the modal without saving changes.

9.6.3 Filter Modal

evalio						Ques	stions	Courses a	and Portfolio	Profile
Portfolio			Filter Cour Select Sem	rses ester:					Add New Course	Filter
Course Name	¢	Course Code	Select Cour	rse Name: urses		~	Actions			
Network		CS421			Cancel	er	Detail	Edit	•	
411.0									Add New Course	Filter
All Courses										
Course Name	¢	Course Code		Department	Semester		Actions			

Figure 66: Image of Filter Modal

This modal appears when the user clicks the Filter button on the Courses and Portfolio page. It allows users to narrow the displayed course list by selecting a specific semester and/or course name. The dropdowns dynamically fetch and list only the active semesters and courses associated with the current user. Once a filter is applied, the course lists below are updated accordingly. The Cancel button closes the modal without applying any changes, while the Filter button triggers the filtering operation.

9.6.4 Course Summary Page

			R.				
Network (S	Spring - 202	25) Portfo	olio	Assign TA / Show Assig	ned TAs Upload /	View Documents	Export Portfolio
Course Summ	nary						
Semester: 202	15 - Spring			Number of Sections	1.3		
A Instructor: Del	fault Instructor	a Annistrat		 Active Course: Co	se is active in this serve	ister	
P. Course Descri	ation: Network is very	upatul		E) Pertine Types: Qui	r, examination, Attend	ance, Plomework	
Add Portfolio Rem	Edit Portfolio Item	0		Dartfolio Itam - Oti	-		
Add Pertfolio Item Portfolio Item - / Examination	Edit Partfalio Item Assignment			Portfolio Item - Oti Attendance	her		
Add Portfolio Rem Portfolio Item - / Examination	Edit Portfolio Item Assignment			Portfolio Item - Oti Attendance	ter		
Add Partfolio Rem Portfolio Item - / Examination Assignment Nar Midtern 5	Edit Portfolio item Assignment Date Assigned 19/04/2025	Average Score 45	Number of Files	Portfolio Item - Otl Attendance	ter escription Itendance data, denem	Numb a D	er of Files
Add Portfolio Rem Portfolio Item - / Examination Assignment Nar Midtern 5 Midtern 1	Edit Portfolio Item Assignment Date Assigned 19/04/2025	Average Score 45 57	Number of Files D 4	Portfolio Item - Ot Attendance	ter escription liendance data, denem	e O	er of Files
Add Perifolio Rem Portfolio Item - / Examination Atalgament Nar Midtern 5 Midtern 1 Midtern 2	Edit Portfolio Item Assignment	Average Score 45 97 73	Number of Files 0 4 1	Portfolio Item - Otl Attendance	her excription tiendance data, denem	e 0	er of Files

Figure 67: Image of Course Summary Page

This page is displayed when the user clicks the Detail button on a course from the Courses and Portfolio page. It provides a structured summary of the selected course and its portfolio items.

At the top, the breadcrumb trail allows users to return to the previous course list page. The heading section displays the course name and semester (e.g., *Network - Spring 2025*). Just beside the title are three primary buttons:

- Assign TA / Show Assigned TAs: used to view or update the TAs responsible for the course.
- Upload / View Documents: navigates to the document upload interface for portfolio items.
- **Export Portfolio**: triggers the download of a compiled PDF portfolio for accreditation or archival.

The first card on the page shows a Course Summary, including the semester, instructor, assigned TAs, number of sections, active status, course description, and all portfolio types registered for that course. This section can be toggled using the eye icon in the card's top-right corner to show or hide details. Below, the page is split into two main sections:

- The left column (Portfolio Item Assignment) lists structured assignment-related portfolio types, such as "Examination" or "Quiz". Within each, assignments like "Midterm 1" or "Quiz 5" are displayed with details including assigned date, average score, and number of uploaded files.
- The right column (Portfolio Item Other) includes non-assignment types such as "Attendance" and "Konuşanlar", organized similarly.

Each item in these tables is clickable. Clicking on an item opens the document management view specific to that item, with uploaded files and details.

At the top of the portfolio section, there are two action buttons:

- Add Portfolio Item: opens a modal to create a new portfolio item under an existing type.
- Edit Portfolio Item: allows modifying or deleting existing portfolio item details if necessary.

The overall layout is designed to give instructors and TAs a quick, editable overview of all portfolio components related to a specific course.

9.6.5 Assign TA modal

🍼 evalio		Questions	Courses and F	Portfolio Profile	G
	Assign Teaching Assistants Search by name or email	Upload	/ View Documents	Export Portfolio	
	Default Teaching Assistant ta@example.com			ø	
Course Summary	TA Deneme yusuf.toraman@ug.bilkent.edu.tr	O		~	
Semester: 2025 - Spring		Cancel Assign			
A Instructor: Default Instructor		 Active Course: Course is active in this ser 	nester		
条 Teaching Assistants: Default Teaching Assista	int	🔁 Portfolio Types: Quiz , Examination, Atter	dance, Homework		
Course Description: Network is very useful					
Add Portfolio Item Edit Portfolio Item					
Portfolio Item - Assignment		Portfolio Item - Other			
Examination		Attendance			

Figure 68: Image of Assign TA Modal

This modal appears on the Course Summary page when the user clicks the Assign TA / Show Assigned TAs button. Instructors can assign or unassign teaching assistants (TAs) to the selected course. The user can search for available TAs by name or email using the input field at the top. Each listed TA has a checkbox, indicating whether they are currently assigned to the course. Ticking or unticking the boxes updates the TA list. Pressing Assign saves the selected TAs, while Cancel closes the modal without applying changes.

9.6.6 Add Portfolio Item Modal

🥑 evalio	Add Portfolio Item	stions Courses and Portfolio Profile G
	* Main Item	
	Select main item v	Ø
Course Summary	* Portfolio Item Name	
Semester: 2025 - Spring	e.g. Homework 4 or Midterm 2	
A Instructor: Default Instructor	* Description	n this semester
条 Teaching Assistants: Default Teaching Assi	Priof description	ion, Attendance, Homework
Course Description: Network is very useful	Bher description	
	Portfolio Item Type	
Add Portfolio Item Edit Portfolio Item	O Assignment O Other	
	Date Assigned	
Portfolio Item - Assignment		
Examination	Average Score	
Assignment Name Date Assigned Ave	Cancel Add	Number of Files
Midterm 5 15/04/2025 45		ata, deneme 0
Midterm 1 14/04/2025 57	4	

Figure 69: Image of Add Portfolio Item Modal

This modal appears when the user clicks the Add Portfolio Item button on the Course Summary page. It allows instructors or teaching assistants to add a new item to the course's portfolio.

Users first select a Main Item (e.g., Examination, Quiz, Attendance) from the dropdown. Then, they provide a Portfolio Item Name (such as "Midterm 2" or "Homework 4"), a brief Description, and specify whether the item is of type Assignment or Other.

If the item is an assignment, users can also optionally enter a Date Assigned and an Average Score field to track performance data. After filling in the form, clicking Add saves the item into the course portfolio, while Cancel dismisses the modal without saving.

9.6.7 Portfolio Files Page

🍼 evalio			Questions	Courses and Portfolio	Profile
Courses > Automata (FALL - 2025) > 1	Portfolio				
Search name, main type or type	Q				Ad
∨ Project					
✓ Project 1					
Title	File	Description	Main Ty	ре Туре	Actions
Project 1 - Best Grade	Best-paper-prj1.pdf	Section 1 - Yusuf Toraman's paper form section 1	Project	Project 1	2 🖸
Project 1 Specification	prj1-specification.pdf	Specification Document	Project	Project 1	2 🖸
Project 1 - Worst Document	Worst-prj1.pdf	Eren Arım's document from section 3	Project	Project 1	2 🗅
> Project 2					
> Quiz					
 Examination 					
> Midterm 1					

Figure 70: Portfolio Files Page

The Portfolio Files Page displays all uploaded portfolio files for a specific course, organized by their Main Type (e.g., Project, Quiz, Examination) and nested Type (e.g., Project 1, Midterm 1). This view is accessible from the Course Summary Page by clicking any portfolio item name.

A breadcrumb trail at the top of the page helps users navigate back to the course context. Directly below, a search bar enables filtering by portfolio name, primary type, or specific type.

Each portfolio item is expandable. When expanded, it reveals a table that lists the following columns:

- Title: A descriptive title for the uploaded file.
- File: A clickable file name that opens the uploaded document.
- **Description**: A brief explanation of the file content or context.
- Main Type: The general category of the portfolio item (e.g., Project).
- **Type**: The subgroup under which the item is categorized (e.g., Project 1).
- Actions: Edit and delete icons that allow users to update metadata or remove the file.

Instructors or TAs can click the Add File button in the top-right corner to add a new file to the portfolio. This opens a modal where the user provides the file, selects the main and specific types, and optionally adds a description. Once saved, the new file will appear under the appropriate group.

Files can be updated or deleted using the corresponding action icons in the table. Any changes made are instantly reflected in the grouped view. This page is designed to help educators maintain an organized, trackable repository of student submissions and key course documents.

9.6.8 Portfolios Page/ Add File Modal

🍼 evalio				Questions	Courses and Portfolio	Profile G
습 Courses > Automata (FALL - 2025) > Po	ortfolio		_			
Search name, main type or type	٩	Add a File * Main Type				Add File
✓ Project		Select main type	×)			
V Project 1		* Sub Type Select sub type	v			
Title	File	* Title		Main Ty	ре Туре	Actions
Project 1 - Best Grade	Best-paper-prj1.pdf	Enter title		Project	Project 1	∠ 0
Project 1 Specification	pri1-specification.pdf	Description		Project	Project 1	2 0
Project 1 - Worst Document	Worst-prj1.pdf	Enter description (optional)		Project	Project 1	2 0
> Project 2						
> Quiz		* File (PDF)				
✓ Examination		Status				
> Midterm 1		Active				
> Final		Section				
			Cancel Add			

Figure 71: Portfolios Page/ Add File Modal

This modal is opened when users click the Add File button on the Portfolios Page. It allows instructors or TAs to upload a PDF document under a selected main and sub portfolio type. Users provide a title, optionally add a description and section info, and mark the file as active or not. Once submitted via the Add button, the file is listed in the appropriate group. Clicking Cancel closes the modal without saving.

9.6.9 Portfolios Page/ Add Portfolio Type Modal

🍼 evalio			Questions	Courses and Portfolio	Profile G
Search name, main type or type	Add a File * Main Type	Add Portfolio Type * Main Type			Add File
✓ Project	Select main type	Select main type V			
V Project 1	* Sub Type Select sub type	* Portfolio Type Name e.g. Hornework 4 or Midterm 2			
Title	* Title	* Description			Actions
Project 1 - Best Grade	Enter title	Brief description	Proje	ect Project 1	20
Project 1 Specification	Description		Proj	ect Project 1	20
Project 1 - Worst Document	Enter description (o	Portfolio Type	Proj	ect Project 1	_ 0
> Project 2		• Assignment Other			
> Quiz	* File (PDF)	Date Assigned			
✓ Examination	Status	Average Score			
> Midterm 1	Active				
> Final	Section	Cancel Add			
		Cancel Add			

Figure 72: Portfolios Page: Add Portfolio Type Modal

"Add Portfolio Type Modal" appears when the "+" button at the bottom of a portfolio type list is clicked after expanding a Main Type or Type section. It allows instructors or TAs to define a new portfolio item category (e.g., "Homework 4" or "Midterm 2") by entering its name, description, and selecting whether it's an assignment or other type. Optionally, the assigned date and average score can be added. Clicking Add creates the new portfolio type, while Cancel closes the modal without changes.

9.6.10 Portfolios Page/ Add Portfolio Type Modal

🧭 evalio			Questions Cours	es and Portfolio	Profile
🖨 Courses > Automata (FALL - 2025) >	Portfolio				
Search name, main type or type	٩	Delete this portfolio? Cancel OK			Add
✓ Project					
✓ Project 1					
Title	File	Description	Main Type	Туре	Actions
Project 1 - Best Grade	Best-paper-prj1.pdf	Section 1 - Yusuf Toraman's paper form section 1	Project	Project 1	∠ ਹ
Project 1 Specification	prj1-specification.pdf	Specification Document	Project	Project 1	_ ਹ
Project 1 - Worst Document	Worst-prj1.pdf	Eren Arim's document from section 3	Project	Project 1	∠ 1
> Project 2					
> Quiz					
✓ Examination					
> Midterm 1					
> Final					

Figure 73: Portfolios Page: Add Portfolio

When the delete icon is clicked for a portfolio file, this confirmation dialog appears. Clicking OK permanently deletes the file, while Cancel aborts the action.

9.6.11 Portfolio Creation Page

	O Courses > Automata (EALL -)	2025) Create Portfolio					
		2020) > Greate Fortiono					
	<pre>170 \hline D & 1.0 & 6 171 \hline F & 0.0 & 6 171 \hline F & 0.0 & 6 172 \hline F & 0.0 & 6 173 \hline Valtcolumn 174 \hline \ulticolumn 175 \hline 175 \hline 176 \vend(tabularx) 178 \vend(tabularx) 179 \vend(tabularx) 179 \vend(tabularx) 179 \vend(tabularx) 179 \vend(tabularx) 170 \vend(tabularx) 170 \vend(tabularx) 170 \vend(tabularx) 170 \vend(tabularx) 170 \vend(tabularx) 171 \vend(tabularx) 172 \vend(tabularx) 173 \vend(tabularx) 173 \vend(tabularx) 174 \vend(tabularx) 175 \vend(tabularx) 17</pre>	<pre>\\ \\ 6 \\ 22\{\ }\Totals} & N= & W= \\ (24\{\ }\Totals} & N= & W= \\ (43\{\}\Average GPA = WV/N} \\ nd Signature Section extwidth}}{(}{}\} (6.6\textwidth) }, \\ bus page at STARS \\ od, one average and one poor \\ tudent work for each exam, project, \\ uiz. \\ the Department at the end of the semester. {0.35\textwidth} \\ \ bus pace between signature and date ace{lem} \underline{\hspace{3cm}} </pre>		1 /1 88 Code: CE 41 Code: 2 Image: Code: CE 41 Code: 2 Code: CE 41 Code: 2 Image: Code: CE 41 Code: 2 Code: CE 41 Code: 2 Image: Code: CE 41 Code: CE 41 CO		Arrive Andrew Andr	din -
	199 \end{document}			C 1 1 Attachments 1) Capy of the split 2) Capy of the split 2) Capy of attachment 4) Capy of attachment 4) Capy of anoma a stant for returned to	also page at STAIS set, one range and sure your set, one range many set, and set, address that and set, many set, address and set of the seconds.	Signature:	
Sea All C	If y vend(document) Irch name, main t Q Documents	•		C 19 C	den ege et ITAB den ege et ITAB fort er ut for est som sport, som er et angenet ut bourteast et be oud of the sensets.	Signature Date	Sav
Sea ([199 Vend{document} Irch name, main t Q Documents Title	Description	Mair	Le 19 International Construction of the second se	Are upper at TAIM to the one to the specific of the second secon	Rgaatere Date: Export	Sav
Sea (II C	199 (end{document}) Inch name, main t Q. Documents Title Project 1 - Best Grade	Description Section 1 - Yusuf Toraman's paper form section 1	Main	C 19 C 19 C 20 C	And ange of IEEM Internet of IEEM Internet of IEEM Internet of IEEM Internet of IEEM Description Section 1 - Yusuf Toran	Spantere Date Export nan's paper form section 1	Sav Main 1 Proje
Sea (II C	199 Vend{document} rrch name, main t Q. Documents Title Project 1 - Best Grade Project 1 Specification	Description Section 1 - Yusuf Toraman's paper form section 1 Specification Document	Main Pr Pr	C C	the upper st TAIM the set of the	Signature: Date: Export Export ann's paper form section 1 t	Sav Main Proje Proje
Sea JII [2]	199 (end(document)) Inch name, main t Q In	Description Section 1 - Yusuf Toraman's paper form section 1 Specification Document Eren Arm's document from section 3	Main Pr Pr Pr	Image: Contract of the second seco	the period of the second of t	Speaterer Date: Export Export an's paper form section 1 t rom section 3	Sav Main Proje Proje
Sea III C	199 (end(document)) rich name, main t Q Documents Title Project 1 - Best Grade Project 1 Specification Project 1 - Worst Document Project 2 - Specification	Description Specification Document for project 2	Mair Pr Pr Pr Pr	Image: Contract of the second secon	the graph of Hand and opport of Hand State with the set of the set state with the set of the set set of the set of the set of the set set of the set of the set of the set set of the set of the set of the set set of the set of the set of the set set of the set of the set of the set set of the set of the set of the set set of the set of the set of the set of the set set of the set of the set of the set of the set set of the set of the set of the set of the set set of the set of the set of the set of the set of the set set of the set of the se	Signature: Bate Export Export ann's paper form section 1 t from section 3 t for project 2	Main Proje Proje Proje
	199 (end(document)) arch name, main t Q bocuments Title Project 1 - Best Grade Project 1 Specification Project 1 Specification Project 2 - Specification Quiz 3	Description Section 1 - Yusuf Toraman's paper form section 1 Specification Document Eren Arm's document form section 3 Specification Document for project 2 Quiz	Mair Pr Pr Pr Pr Qu	Image: Contract of the second seco	the provide a set of the set of	Speaker: Date: Export Export for section 3 t for project 2	Main Proje Proje Proje Quiz
	199 (end{document}) Inch name, main t Q. Inch name, main t Q. Documents Title Project 1 - Best Grade Project 1 - Specification Project 2 - Specification Quiz 3 Midterm 1 - Solution Key		Mair Pr Pr Pr Pr Qu Ex	Image: Contract of the second seco	Internet and the set of the set o	Signature: Date: Export Export t from section 3 t for project 2	Main Proje Proje Proje Quiz Exan
Sea 11 (2) 2) 2) 2) 2) 2) 2)	199 (end(document)) arch name, main t Q bocuments Title Project 1 - Best Grade Project 1 - Specification Project 2 - Specification Quiz 3 Midterm 1 - Solution Key Final	Description Section 1 - Yusuf Toraman's paper form section 1 Specification Document Specification Document for project 2 Specification Document for project 2 Quiz Solution key Final Exam Paper (Empty)	Mair Pr Pr Pr Qu Ex Ex	Image: Control of Contro	the menung and stars project where we have a star project the menung and the star and the star the menung and the star and the star star and th	Sepontaries Date: Export Export for project 2 fy)	Sav Main Proje Proje Proje Quiz Exan Exan

Figure 74: Portfolio Creation Page

The Portfolio Creation Page is used to generate a complete course portfolio in PDF format. On the left side, an automatically generated LaTeX summary is provided, which can be edited manually if needed. As users make changes, the compiled PDF version is rendered on the right side for real-time preview.

Below the editor, all uploaded documents are listed in the All Documents panel on the left. Clicking on any document and then pressing the Save button adds it to the Selected Documents panel on the right. These selected files will be included in the final output.

When the Export button is clicked, the manually edited LaTeX summary and the selected portfolio files are merged and downloaded together as a single, comprehensive PDF portfolio. This allows instructors and TAs to create a ready-to-submit report for accreditation or archiving purposes.

10. Glossary

ABET / MÜDEK: Accreditation organizations that assess the quality of engineering programs. Evalio supports generating reports that are aligned with their standards.

Accreditation Reports: Reports generated to demonstrate compliance with educational standards like ABET and MÜDEK.

Admin: The highest-privileged system user responsible for approving institutions and overseeing platform-wide data.

API Gateway: A central entry point that routes and manages client requests to backend microservices.

Assignment Component: A grading item (e.g., Quiz, Homework) within a course's evaluation structure.

Authentication Service: A microservice responsible for verifying user credentials and enforcing RBAC.

Auto Scaling: A system mechanism that adjusts resources automatically based on traffic or load.

Cloud Storage (AWS S3): A cloud-based storage for storing documents and files externally, especially in the Portfolio module.

Course Portfolio: A structured archive of course materials such as exams, assignments, and attendance records.

Docker: A containerization platform used to deploy Evalio's microservices consistently across environments.

Exam Builder: A module that allows instructors to create and manage exams by selecting and organizing questions.

JWT (JSON Web Token): A token used for user authentication and securely transmitting role and identity data.

LaTeX: A document formatting language used to write and render math-heavy exam content professionally.

LaTeX Compiler: A tool integrated into Evalio to convert LaTeX content into rendered PDF output.

Microservice: A self-contained, independently deployable service responsible for a specific functionality.

Microservices Architecture: A design pattern in which different services handle specific business logic independently.

Objection Session: A time window where students can review graded exams and optionally raise objections.

PDF Export: The process of generating a downloadable PDF version of exams, questions, or portfolios.

Portfolio Builder: A module that allows instructors to upload, organize, and export course-related documents.

Portfolio Item: A document or academic artifact (e.g., exam paper, attendance sheet) within a course portfolio.

Redis: An in-memory key-value store used for caching and real-time session management.

Redis Cache: The caching layer of Evalio that improves performance by reducing load on the database.

RBAC (Role-Based Access Control): A security mechanism that restricts access to features based on user roles.

Service Discovery: The mechanism by which microservices locate each other within the deployment environment.

Slot: A reservable time unit within an objection session for students.

TA (Teaching Assistant): A user role with permissions to manage objection sessions and upload portfolio materials.

TexLive: The LaTeX distribution used to compile LaTeX source code into PDF documents in Evalio.

Two-Factor Authentication (2FA): A security feature requiring a password and a temporary verification code.

UML (Unified Modeling Language): A standard way of visually representing system architecture and workflows.

User Roles: Predefined access levels in Evalio, such as Instructor, Teaching Assistant, Institution, and Admin.
11. References

[1] Purdue University, "Creating Exams," Purdue Innovative Learning, 2024. [Online]. Available:

https://www.purdue.edu/innovativelearning/teaching/module/creating-exams/. [Accessed: Nov. 21, 2024].

- [2] Sourab Pramanik, "Deploy your React App using Docker and Nginx," DEV Community, Nov. 27, 2023. https://dev.to/sourabpramanik/deploy-your-react-app-using-docker-and-nginx-14lk ?utm_source=chatgpt.com (accessed May 01, 2025).
- [3] "Consul Documentation | Consul | HashiCorp Developer." Consul Documentation | Consul | HashiCorp Developer, developer.hashicorp.com/consul/docs.
- [4] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," May 2015, doi: https://doi.org/10.17487/rfc7519.

[5] "Autoscaling Workloads." Kubernetes.io, kubernetes.io/docs/concepts/workloads/autoscaling/.