SE 342

Software Validation and Testing

REAL TIME ATTENDANCE SYSTEM

22 07 06 010- Taylan Alp Çakı

22 07 06 011-Erdem Beler

22 07 06 034-Burçak Çelt

22 07 06 038-Ecem Nur Özer

REAL-TIME ATTENDANCE SYSTEM

1. Introduction

The **Real-Time Attendance System** is designed to automate the process of student attendance monitoring in classrooms using **real-time facial recognition technology**. The system aims to improve accuracy, save time for instructors, and provide detailed reporting and notifications for both students and administrators.

The project is developed using **Scrum methodology**, with work organized into **epics** and **stories/tasks** tracked in **Jira**. This report provides an overview of the system, the structure of epics and stories, and their short descriptions.

2. Project Objectives

1. Automate attendance tracking in classrooms with minimal human intervention.
2. Implement real-time facial recognition with high accuracy.
3. Provide reporting and notification functionality for students and administrators.
4. Ensure system security, scalability, and efficient performance.
5. Maintain a modular and extensible architecture for future improvements.

3. System Architecture Overview

The system is composed of the following modules:

- **User Interface (UI):** Web-based interfaces for administrators and students.
- **Backend Services:** APIs, business logic, and integration with the database.
- **AI and Image Processing:** Real-time face detection and recognition.
- **Database & Data Structures:** Storage of attendance records and user information.
- **Notifications & Reporting:** Automated alerts, summary reports, and dashboards.

4. Epics and Stories

Epic 1 – System Requirements and Analysis

This epic covers understanding user requirements, defining system specifications, and preparing initial designs.

Stories under Epic 1:

1. **Administrator and User Interfaces** – Design UI for admin and student dashboards.
2. **Reporting and Notification Functionality** – Plan report generation and notification mechanisms.
3. **Core Attendance and Security Rules** – Define attendance rules and security policies.
4. **Gather Functional Requirements from Stakeholders** – Collect requirements from instructors and students.
5. **Define Non-Functional Requirements** – Specify performance, reliability, and security criteria.
6. **Create Use Case Diagrams and User Journeys** – Visualize workflows and system interactions.
7. **System Architecture Overview & High-Level Design** – Prepare initial system architecture sketches.
8. **Risk Assessment and Feasibility Documentation** – Identify potential risks and plan mitigation strategies.

Epic 2 – Analysis & Technical Design

This epic focuses on technical design, including software architecture, database structure, and AI model planning.

Stories under Epic 2:

1. **Hardware and Infrastructure Planning** – Define hardware requirements and network setup.
2. **AI Model and Image Processing Architecture** – Design AI model and image processing workflow.
3. **Database Architecture and Data Structure** – Plan database schema and data organization.
4. **Backend Service and Data Flow Architecture** – Define backend services and data flow.
5. **API Endpoint Specifications** – Specify endpoints for communication between frontend and backend.
6. **System Security & Encryption Design** – Plan security measures and encryption methods.
7. **UI/UX Wireframes for Dashboard** – Create wireframes for dashboards and reporting views.

8. **Load Handling and Scalability Planning** – Ensure the system can handle high loads efficiently.

Epic 3 – Real-Time Attendance Processing

This epic involves implementation of the core attendance functionality, including video capture, recognition, and data integration.

Stories under Epic 3:

1. **Video Capture & Stream** – Implement video streaming from classroom cameras.
2. **Recognition & Matching** – Process captured frames to recognize students' faces.
3. **Attendance Record Creation** – Save attendance records in the database.
4. **Real-Time Event Triggering** – Trigger events when attendance is detected.
5. **False Positive & Negative Handling Logic** – Handle recognition errors and exceptions.
6. **Performance Optimization** – Optimize recognition speed and system response time.
7. **Failure Logging & Monitoring System** – Log errors and monitor system performance.
8. **Integration with Backend API** – Connect recognition system with backend services for reporting.

5. Project Management

- **Methodology:** Scrum
- **Project Tracking Tool:** Jira
- **Issue Types:** Epics, Stories, Tasks
- **Sprint Planning:** Tasks are planned and estimated with story points to track progress.
- **Version Control:** Git (for backend and AI modules)

6. Conclusion

The Real-Time Attendance System provides a modular, scalable, and automated solution for monitoring classroom attendance. By using Scrum methodology and structured Jira tracking, the project ensures **progress transparency** and **efficient management** of tasks. The system is designed for future extensibility, allowing integration of new features and improvements over time.

Proje: Otomatik Yüz Tanıma Tabanlı Yoklama Sistemi Geliştirilmesi

1. Ana Hedef (Epic)

- **Epic Adı:** Otomatik Yüz Tanıma Tabanlı Yoklama Sistemi Geliştirilmesi
- **Açıklama:** Sınıf kamerasından alınan görüntülerle öğrenci tanıma yaparak yoklama alma ve raporlama sürecini otomatikleştirmek.

2. Temel Bileşenler ve Görevler (User Stories ve Tasks)

İstenen sadeleştirme ve detaylandırma ile temel bileşenleri ve alt görevleri (Tasks) aşağıda listeleyebiliriz:

A Seçeneği (Tek Seçenek: Öğrenci Kaydı, Veri Yönetimi ve Raporlama)

Bu bölümde, yazılımsal altyapı ve veri yönetimine odaklanıyoruz.

Jira Öğesi	Adı	Açıklama
User Story	**Öğrenci Veri Yönetimi ve Yoklama Kaydı**	İdari personelin öğrenci verilerini yönetebilmesi ve sistemin yoklama sonuçlarını doğru bir şekilde kaydedebilmesi.
Task	Öğrenci Veritabanı (DB) Tasarımı	Öğrenci ID, İsim, Sınıf, Yüz Verisi (Encoding/Vektör) gibi alanları içeren veritabanı şemasını tasarla.
Task	Yönetici Arayüzü (CRUD) Geliştirme	İdari personelin yeni öğrenci ekleyebileceği, mevcut öğrencileri güncelleyebileceği ve silebileceği (CRUD işlemleri) bir arayüz geliştir.
Task	Yoklama Sonuçlarını Kaydetme Mekanizması	Tanıma sistemi tarafından gelen veriyi (Öğrenci ID, Ders ID, Saat, Tanıma Başarısı) veritabanına kaydedecek servisi yaz.
Task	Devamsızlık ve Katılım Raporları Oluşturma	Öğretmenin, belirli bir ders/tarih aralığı için devamsızlık ve katılım istatistiklerini görebileceği raporlama modülünü geliştir.

B Seçeneği (Kamera Kurulumu ve Teknik Altyapı)

Bu bölümde, kamera ve donanımsal kurulum ile yapay zeka/yazılım entegrasyonuna odaklanıyoruz.

Jira Öğesi	Adı	Açıklama
User Story	**Sınıf Ortamı Entegrasyonu ve Tanıma Modülü**	Kameranın sınıf ortamına fiziksel kurulumu ve canlı görüntü akışını analiz edecek yapay zeka modelinin entegrasyonu.
Task	**Kamera ve Montaj Planlaması**	Sınıf planına uygun kamera modelini (örn. geniş açılı IP kamera) seç ve montaj (yükseklik, açı, aydınlatma) planını çıkar.
Task	Kamera Montajı ve Kablolama	Seçilen kameraların sınıflara fiziksel olarak monte edilmesi, Cat6 kablolamasının ve güç bağlantılarının yapılması.
Task	Görüntü Akışı (Video Streaming) Servisi Kurulumu	Kameradan alınan canlı video verisini yapay zeka modülüne iletecek *stream* servisinin (örn. RTSP/WebRTC) kurulumu.
Task	Yüz Tanıma Modelini Eğitme	Kullanılacak yapay zeka modelinin (örn. ResNet tabanlı bir Face Recognition modeli) referans veri setiyle eğitilmesi ve optimizasyonu.
Task	Canlı Tanıma ve Eşleştirme Modülü Geliştirme	Görüntü akışında yüzleri algılayacak, tanıyacak ve tanınan öğrencilerin ID'lerini çıktı olarak verecek yazılım modülünü geliştir.
Task	Başarılı Tanıma Bildirimi	Tanıma işlemi başarılı olduğunda, sonuçların "A Seçeneğindeki Kayıt Mekanizmasına" gönderilmesi ve bu entegrasyonun test edilmesi.

Bu yapı ile hem işin yazılım (A Seçeneği) hem de donanım/teknik entegrasyon (B Seçeneği) kısımlarını ayrı iş akışları olarak takip edebilirsiniz.

Jira Öğesi	Adı	Açıklama	Bağlantılı User Story/Task'ler
User Story	**Yoklama Kayıtlarının Kalıcı Hale Getirilmesi**	Sistem, tanınan ve tanınmayan öğrencilerin bilgilerini ders bazında güvenli ve kalıcı bir şekilde depolayabilmelidir.	**B Seçeneği:** Başarılı Tanıma Bildirimi
Task	Yoklama Veri Yapısı Tasarımı	Yoklama kaydının içerik yapısını (Ders ID, Öğrenci ID, Tarih, Saat, Durum: Var/Yok) veritabanı açısından optimize et.	**A Seçeneği:** Öğrenci Veritabanı (DB) Tasarımı
Task	Veri Güvenliği ve Erişim Kontrolü	Yoklama kayıtlarına yetkisiz erişimi önlemek için güvenlik protokollerini (rol tabanlı erişim) tanımla ve uygula.	**Genel Proje Güvenliği**
User Story	**Öğretmen Yoklama Görüntüleme Ekranı**	Öğretmenler, girdikleri dersin yoklama sonuçlarını (Gelenler/Gelmyenler) anlık olarak görebilmelidir.	**A Seçeneği:** Yönetici Arayüzü Geliştirme
Task	Ön Yüz (Frontend) Geliştirme	Öğretmenlerin ders seçimi yapabileceği ve ilgili dersin katılım listesini görebileceği arayüzü kodla.	**A Seçeneği:** Yönetici Arayüzü Geliştirme
Task	Yoklama Veri Çekme API'si	Ön yüze anlık yoklama verisini sağlayacak, filtreleme özellikli (ders/tarih) API servisini geliştir.	**A Seçeneği:** Yoklama Sonuçlarını Kaydetme Mekanizması
User Story	**Detaylı Devamsızlık Raporlama Modülü**	İdari personel, tüm sınıflar ve dersler için belirli bir dönemdeki öğrenci devamsızlık oranlarını ve istatistiklerini raporlayabilmelidir.	**A Seçeneği:** Devamsızlık ve Katılım Raporları Oluşturma

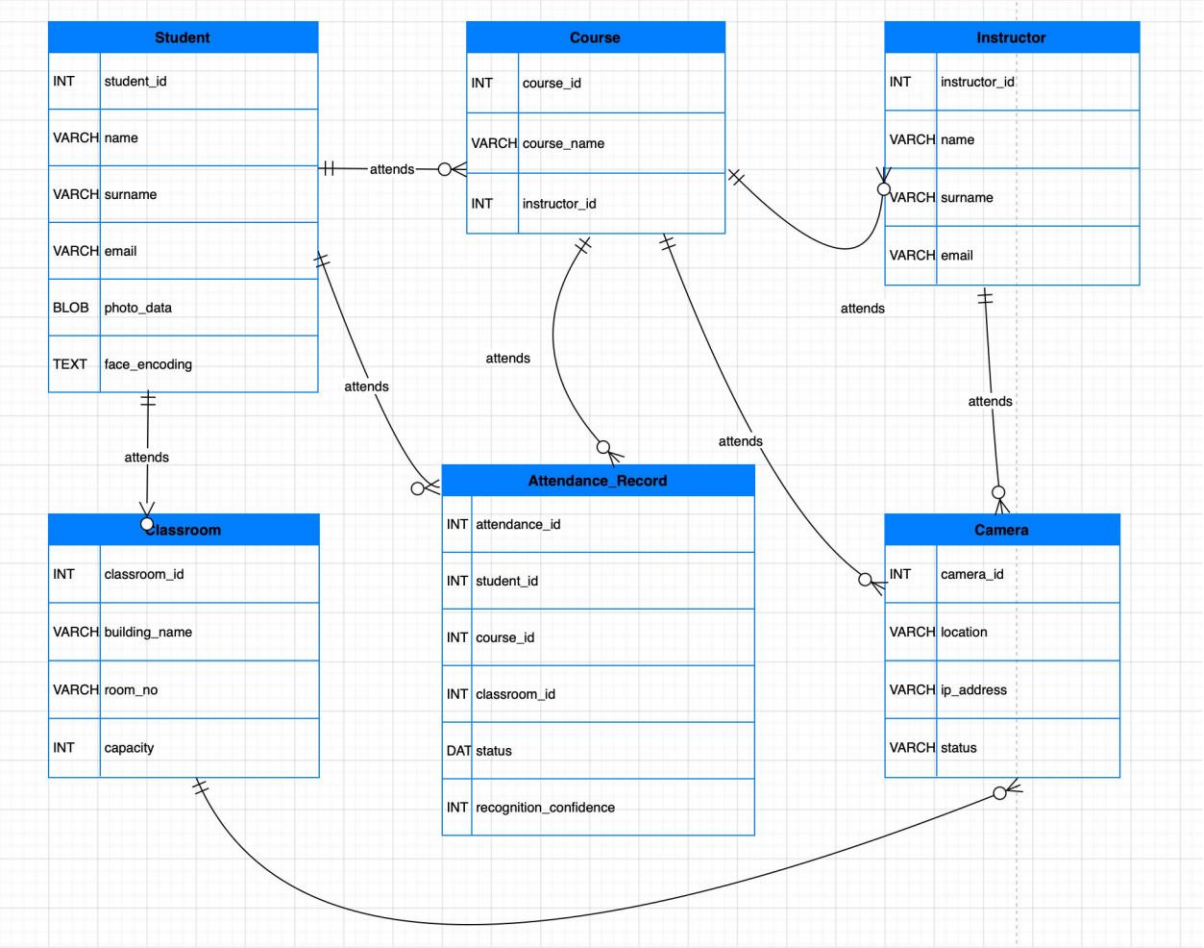
Task	Toplu Veri Analiz Motoru Geliştirme	Belirlenen filtreler (sınıf, dönem, ders) bazında devamsızlık yüzdelerini hesaplayacak arka plan analiz motorunu yaz.	**A Seçeneği:** Devamsızlık ve Katılım Raporları Oluşturma
Task	Rapor İndirme İşlevi (Export)	Oluşturulan raporları PDF, CSV veya Excel formatında dışa aktarma (export) işlevini ekle.	**A Seçeneği:** Devamsızlık ve Katılım Raporları Oluşturma

The Jira Board Link: https://burcakcelt.atlassian.net/jira/software/projects/SCRUM/list

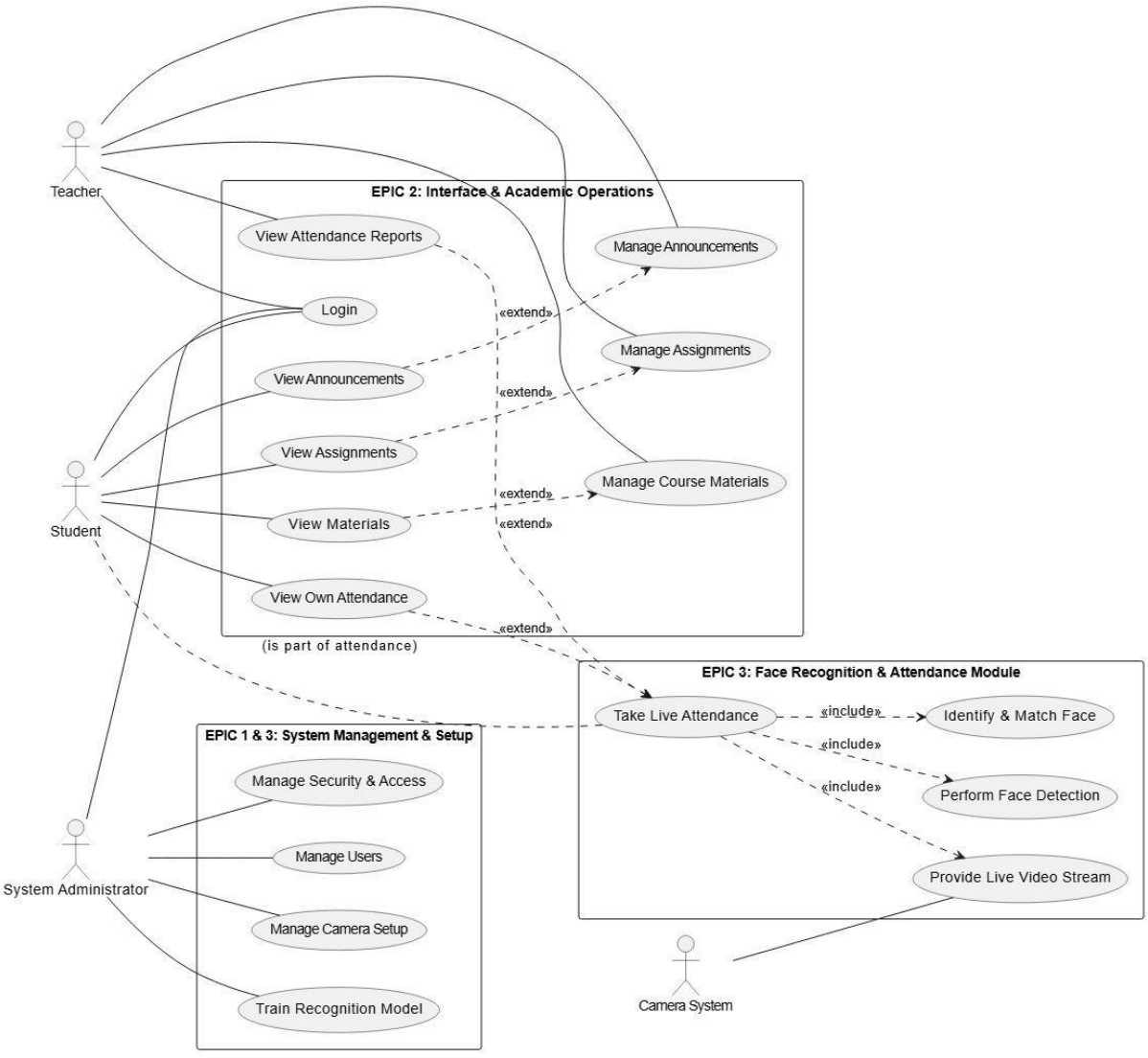
The GitHub Link: https://github.com/Taylan361/real-time-attendence

The app Link: https://real-time-attendence.vercel.app/

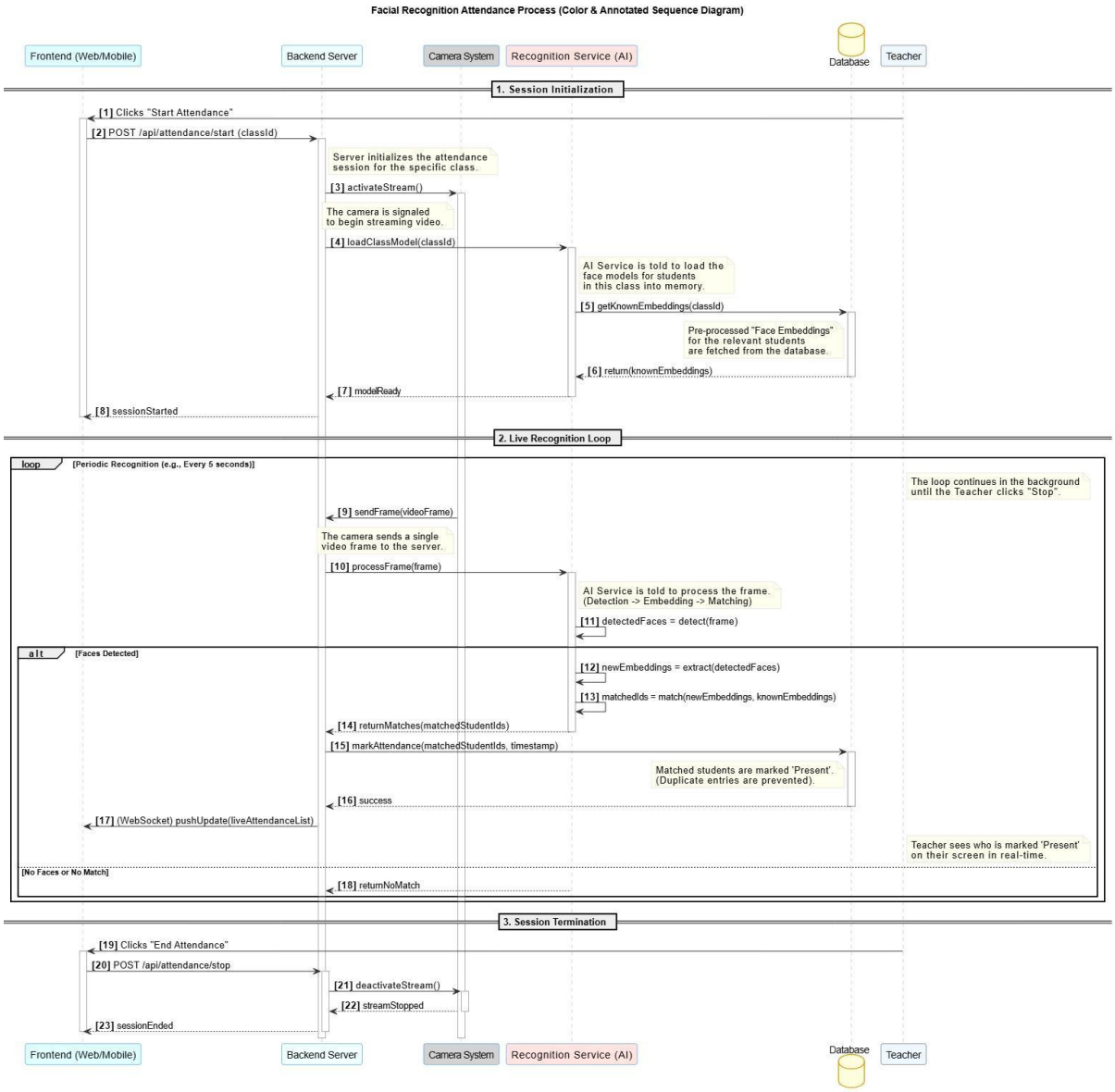
ER DIAGRAM


Student
INT student_id
VARCH name
VARCH surname
VARCH email
BLOB photo_data
TEXT face_encoding
Course
INT course_id
VARCH course_name
INT instructor_id
Instructor
INT instructor_id
VARCH name
VARCH surname
VARCH email
Classroom
INT classroom_id
VARCH building_name
VARCH room_no
INT capacity
Attendance_Record
INT attendance_id
INT student_id
INT course_id
INT classroom_id
DAT status
INT recognition_confidence
Camera
INT camera_id
VARCH location
VARCH ip_address
VARCH status
attends
attends
attends
attends
attends
attends
attends
attends

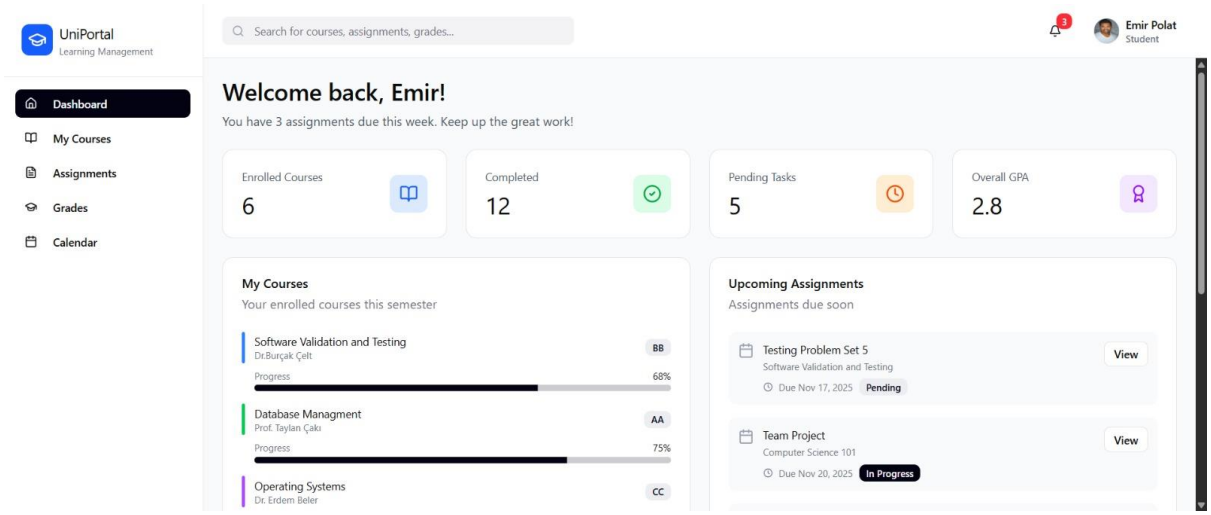
USE CASE DIAGRAM

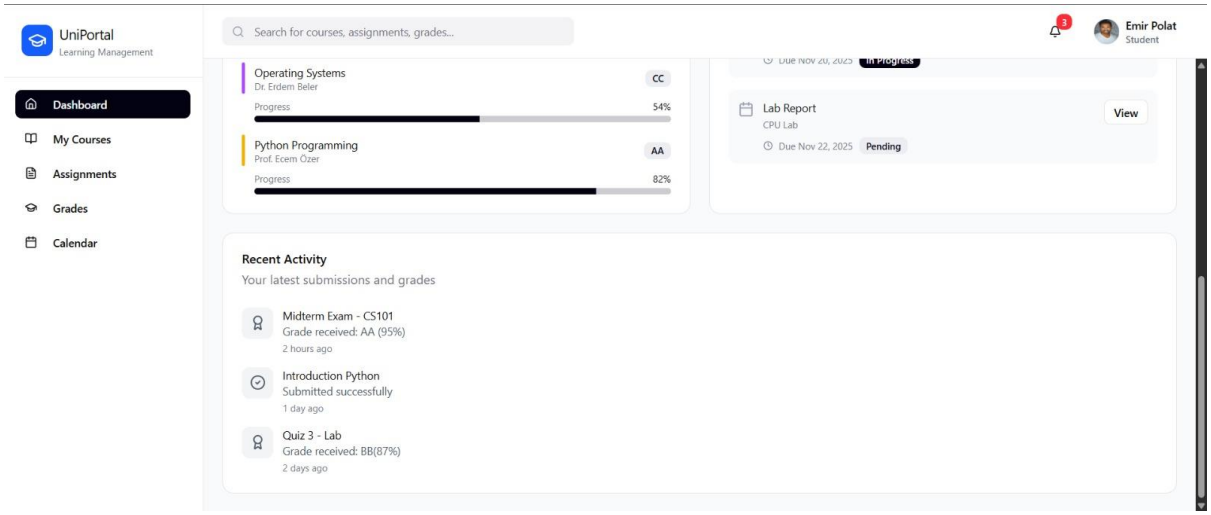


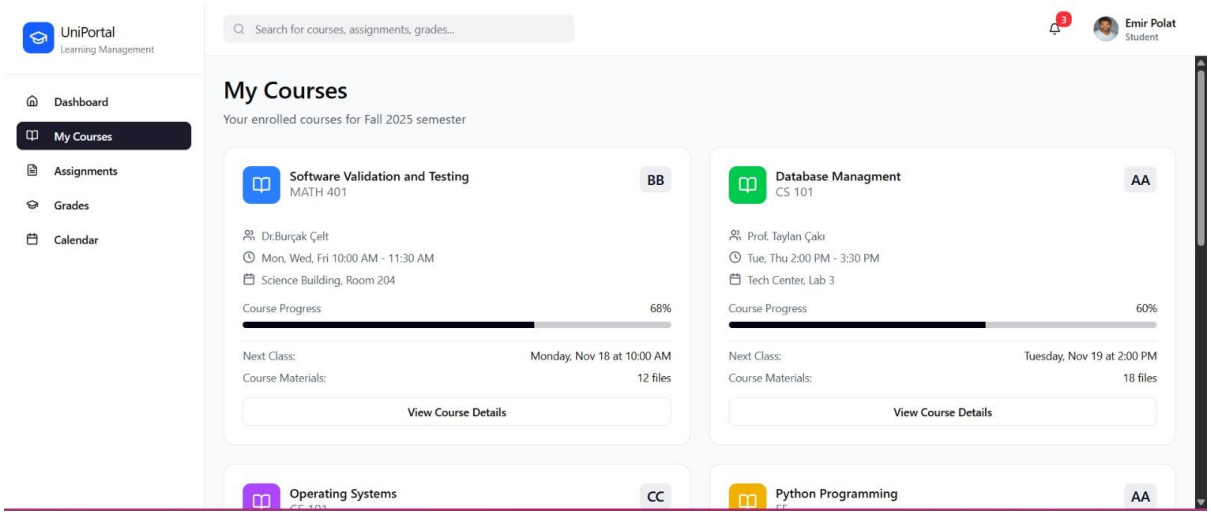
SEQUENCE DIAGRAM

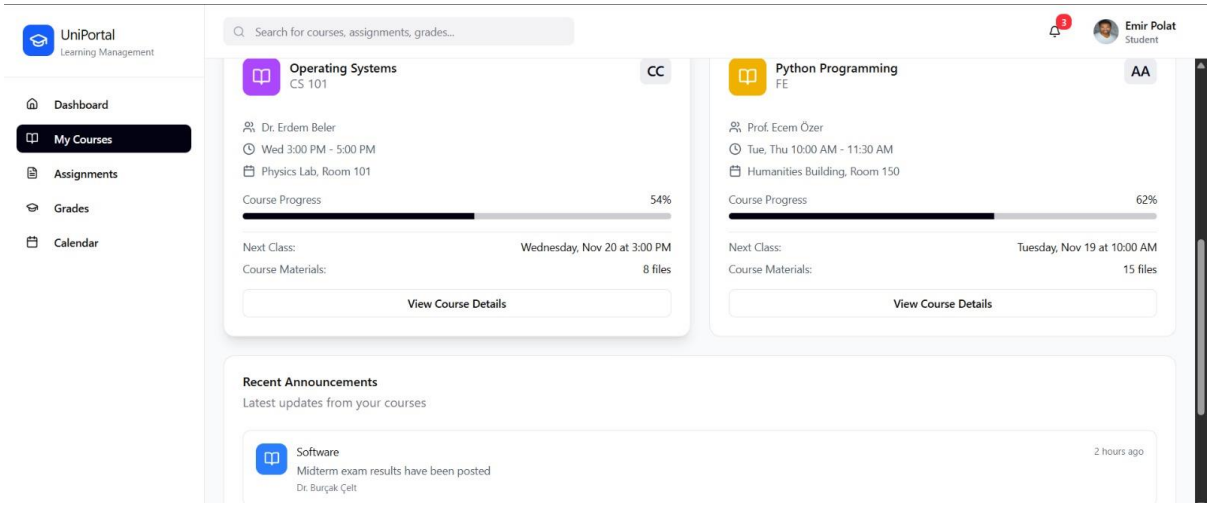


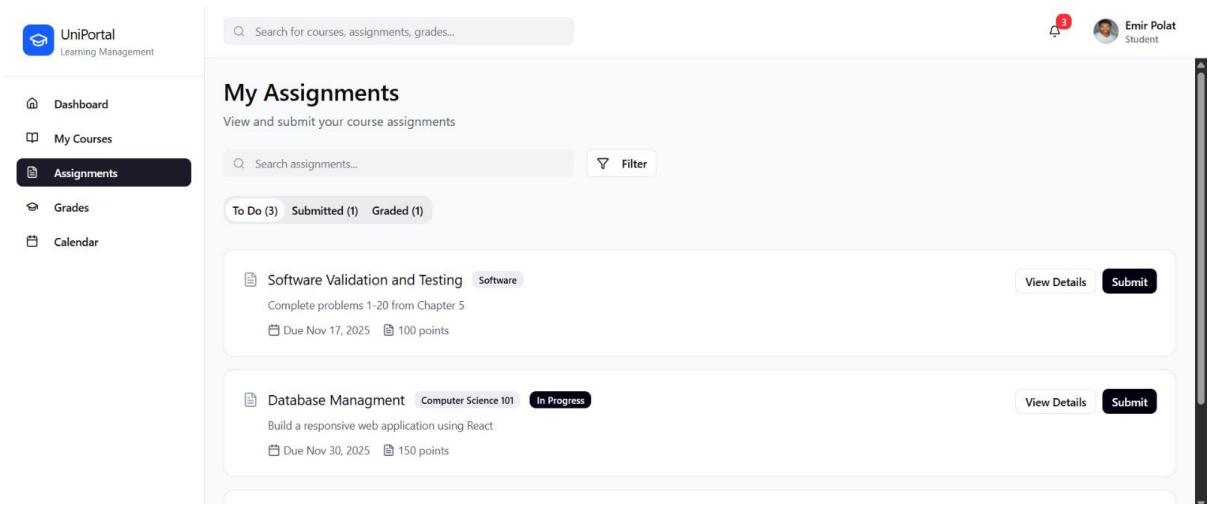
STUDENT INTERFACE FIGMA DESIGN

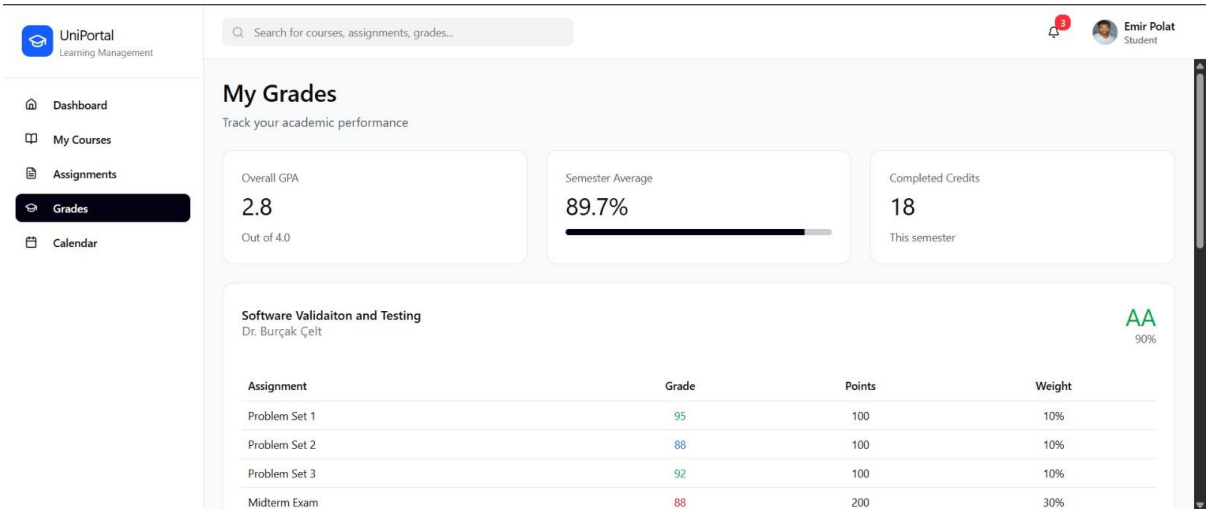


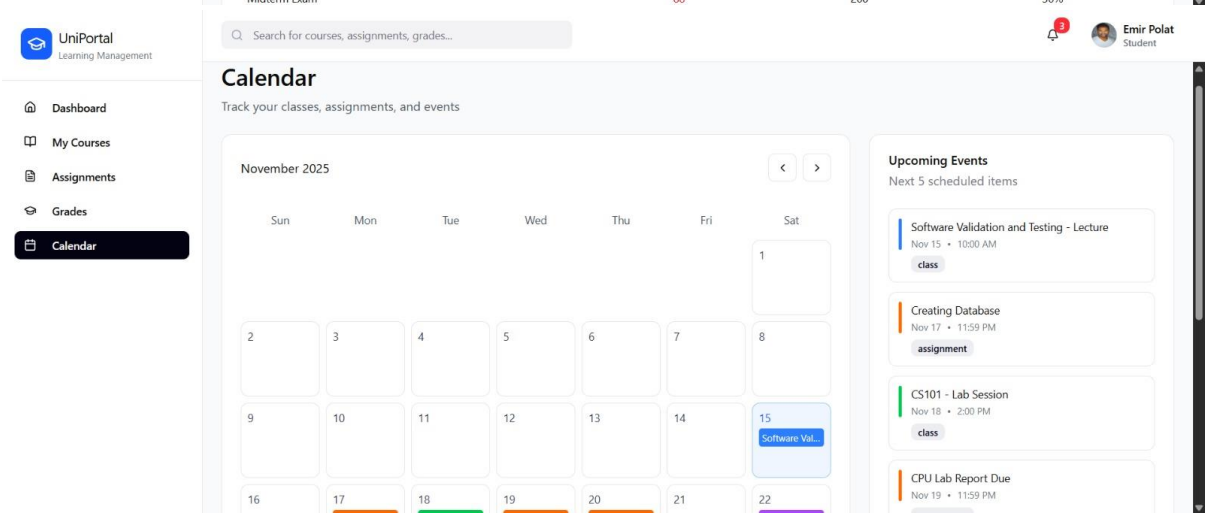




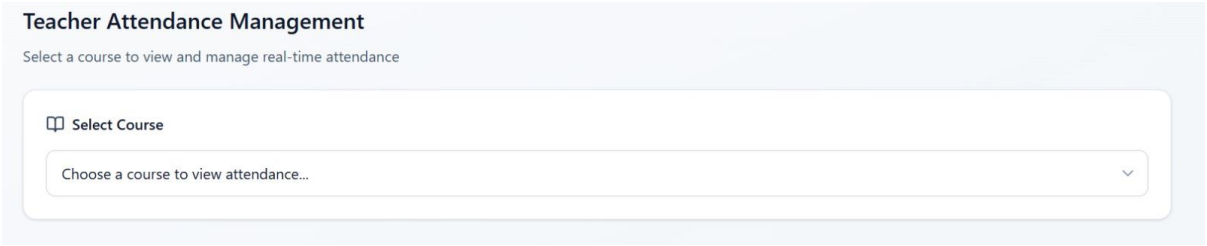

UniPortal
Learning Management
Search for courses, assignments, grades...
3
Emir Polat
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Operating Systems
CS 101
CC
Dr. Erdem Beler
Wed 3:00 PM - 5:00 PM
Physics Lab, Room 101
Course Progress
54%
Next Class:
Wednesday, Nov 20 at 3:00 PM
Course Materials:
8 files
View Course Details
Python Programming
FE
AA
Prof. Ecem Özer
Tue, Thu 10:00 AM - 11:30 AM
Humanities Building, Room 150
Course Progress
62%
Next Class:
Tuesday, Nov 19 at 10:00 AM
Course Materials:
15 files
View Course Details
Recent Announcements
Latest updates from your courses
Software
Midterm exam results have been posted
Dr. Burçak Çelt
2 hours ago

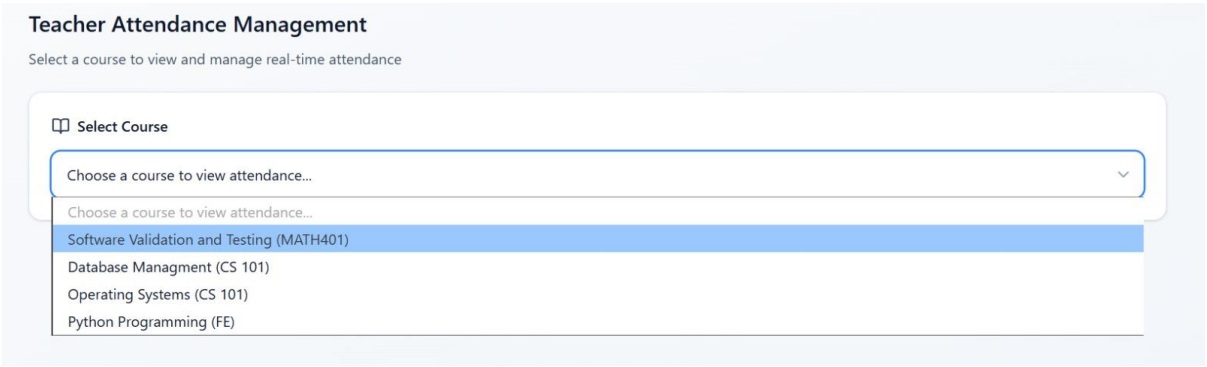

UniPortal
Learning Management
Search for courses, assignments, grades...
3
Emir Polat
Student
Dashboard
My Courses
Assignments
Grades
Calendar
My Assignments
View and submit your course assignments
Search assignments...
Filter
To Do (3)
Submitted (1)
Graded (1)
Software Validation and Testing
Software
Complete problems 1-20 from Chapter 5
Due Nov 17, 2025
100 points
View Details
Submit
Database Managment
Computer Science 101
In Progress
Build a responsive web application using React
Due Nov 30, 2025
150 points
View Details
Submit

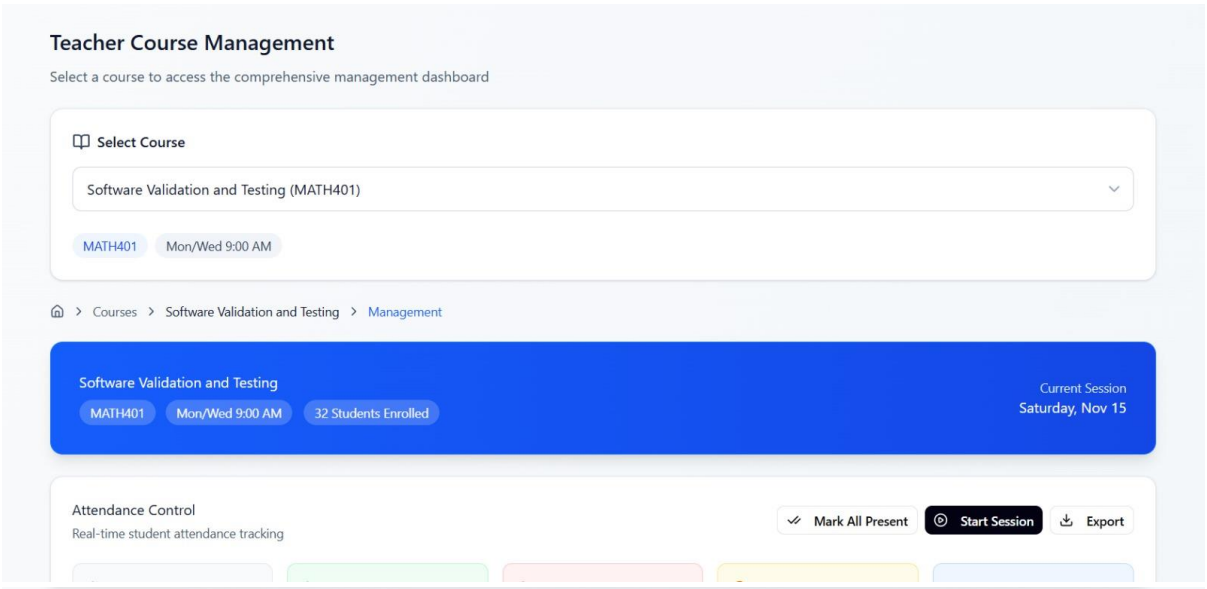

UniPortal
Learning Management
Search for courses, assignments, grades...
3
Emir Polat
Student
Dashboard
My Courses
Assignments
Grades
Calendar
My Grades
Track your academic performance
Overall GPA
2.8
Out of 4.0
Semester Average
89.7%
Completed Credits
18
This semester
Software Validaiton and Testing
Dr. Burçak Çelt
AA
90%
Assignment
Grade
Points
Weight
Problem Set 1
95
100
10%
Problem Set 2
88
100
10%
Problem Set 3
92
100
10%
Midterm Exam
88
200
30%

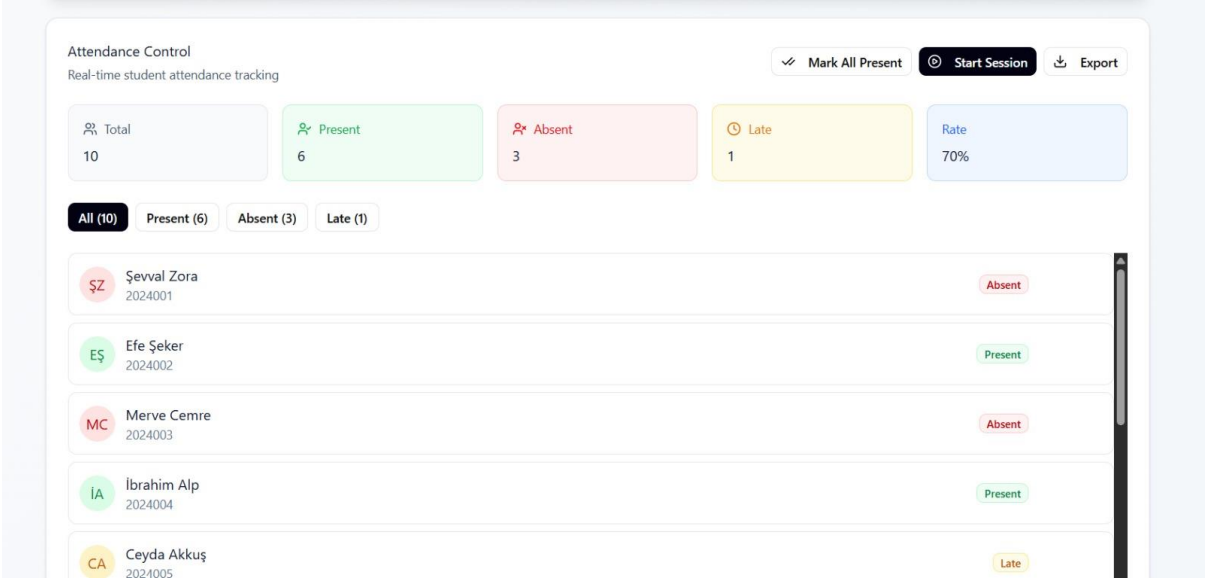

UniPortal
Learning Management
Search for courses, assignments, grades...
3
Emir Polat
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Calendar
Track your classes, assignments, and events
November 2025
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Software Val...
Upcoming Events
Next 5 scheduled items
Software Validation and Testing - Lecture
Nov 15 • 10:00 AM
class
Creating Database
Nov 17 • 11:59 PM
assignment
CS101 - Lab Session
Nov 18 • 2:00 PM
class
CPU Lab Report Due
Nov 19 • 11:59 PM

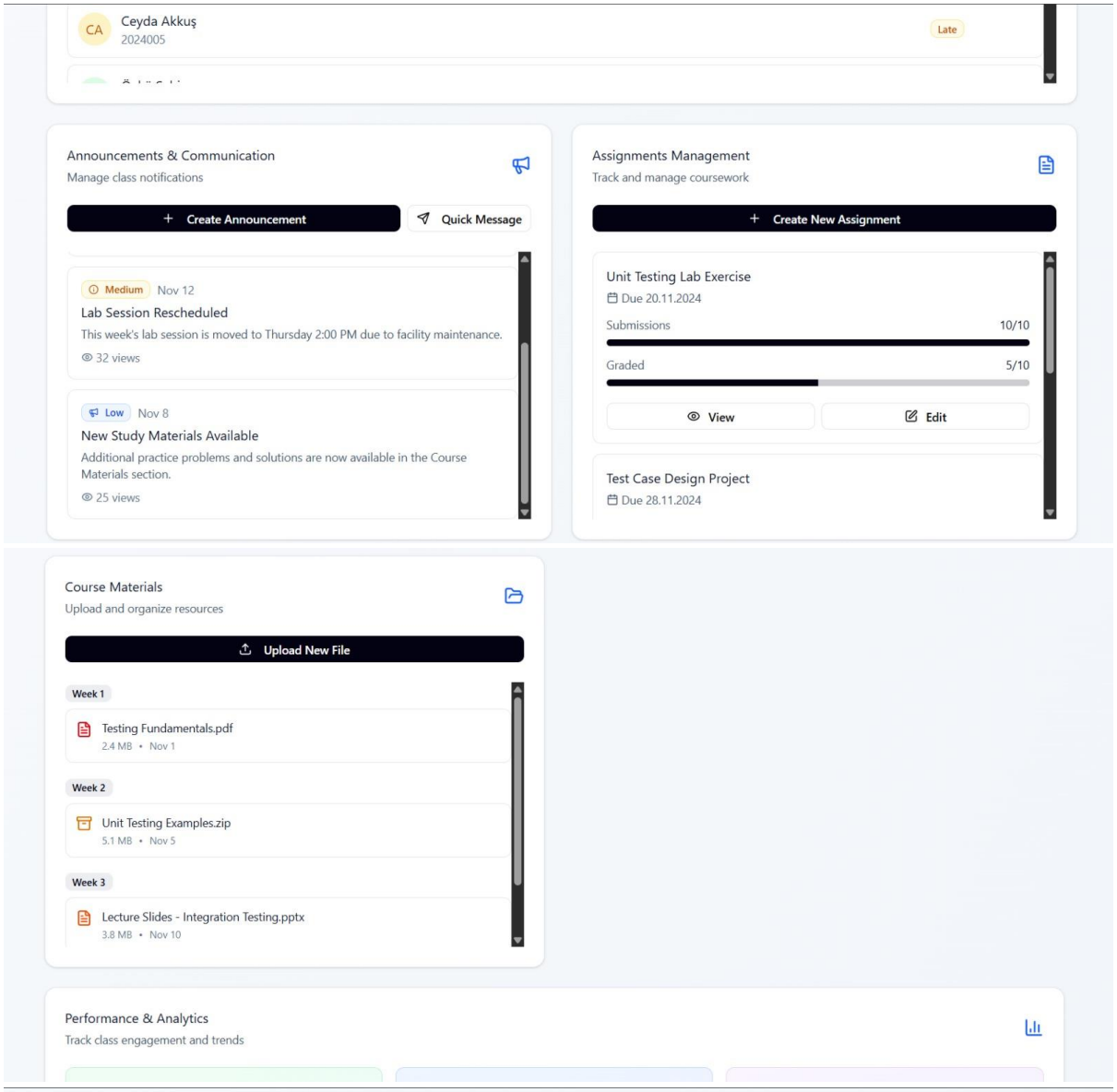
TEACHER INTERFACE FIGMA DESIGN

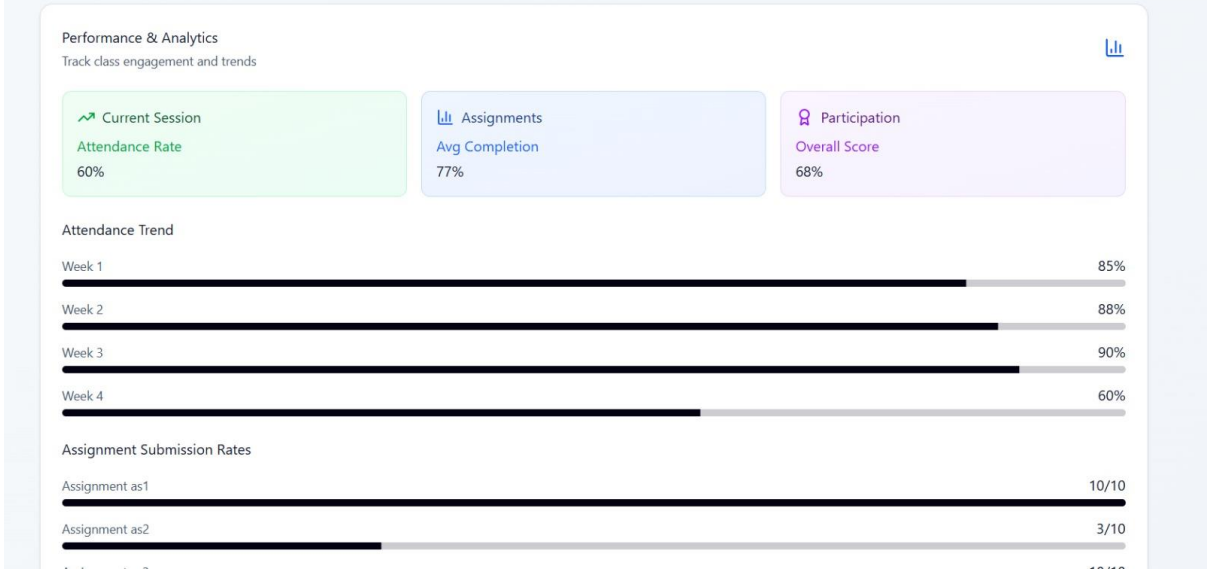









CA
Ceyda Akkuş
2024005
Late
Announcements & Communication
Manage class notifications
Create Announcement
Quick Message
Medium
Nov 12
Lab Session Rescheduled
This week's lab session is moved to Thursday 2:00 PM due to facility maintenance.
32 views
Low
Nov 8
New Study Materials Available
Additional practice problems and solutions are now available in the Course Materials section.
25 views
Assignments Management
Track and manage coursework
Create New Assignment
Unit Testing Lab Exercise
Due 20.11.2024
Submissions
10/10
Graded
5/10
View
Edit
Test Case Design Project
Due 28.11.2024
Course Materials
Upload and organize resources
Upload New File
Week 1
Testing Fundamentals.pdf
2.4 MB • Nov 1
Week 2
Unit Testing Examples.zip
5.1 MB • Nov 5
Week 3
Lecture Slides - Integration Testing.pptx
3.8 MB • Nov 10
Performance & Analytics
Track class engagement and trends


Performance & Analytics
Track class engagement and trends
Current Session
Attendance Rate
60%
Assignments
Avg Completion
77%
Participation
Overall Score
68%
Attendance Trend
Week 1
85%
Week 2
88%
Week 3
90%
Week 4
60%
Assignment Submission Rates
Assignment as1
10/10
Assignment as2
3/10

TESTING TABLES

Test ID	Test Case Description	Steps to Reproduce	Expected Result	
STU-01	**Verify Student Login & Dashboard Data**	1. Open Login Page. 2. Enter Student ID (e.g., 220706010) and Password.	Dashboard loads successfully. The student's name (e.g., "Öykü") and correct stats (GPA, Pending Tasks) are displayed.	**PASS**
STU-02	**Verify Assignment Filtering (Tabs)**	1. Navigate to "Assignments" page. 2. Click on "To Do" tab. 3. Click on "Submitted" tab.	The list below the tabs updates dynamically. "To Do" shows pending items, "Submitted" shows completed ones.	**PASS**
STU-03	**Verify Assignment File Upload Simulation**	1. Go to an assignment detail page. 2. Click on the "Upload Zone" area. 3. Select a file from the computer.	The button changes to "Uploading...", waits 2 seconds, shows a green "Success" message, and redirects back to the list.	**PASS**
STU-04	**Verify Navigation to Calendar**	1. On the Sidebar, click "Calendar". 2. Click on a date with an event (e.g., Nov 18).	The Calendar view loads. Clicking a date updates the "Agenda" card on the right with that day's specific events.	**PASS**
STU-05	**Verify "View All" Buttons Functionality**	1. Go to the Dashboard (Home). 2. Find the "My Courses" section card. 3. Click the "View All Courses" button.	The application redirects the user to the MyCourses page where all enrolled courses are listed.	**PASS**

Test ID	Test Case Description	Steps to Reproduce	Expected Result	
CRS-01	**Verify Course Details Navigation**	1. Go to "My Courses" page. 2. Click "View Course Details" on "Software Validation".	The CourseDetails page opens. The header title displays "Software Validation and Testing".	**PASS**
CRS-02	**Verify Syllabus List Rendering**	1. Open a Course Detail page. 2. Scroll to the "Syllabus" section.	The weekly topics (Week 1, Week 2, etc.) are listed. Completed weeks have a green border/indicator.	**PASS**
CRS-03	**Verify Material Download Buttons**	1. Open a Course Detail page. 2. Locate the "Course Materials" section.	Since it's a demo, the button should be clickable and visually react (hover effect). (In a real app, a download starts).	**PASS**
CRS-04	**Verify "Back" Button Functionality**	1. Open a Course Detail page. 2. Click the "← Back to Courses" button at the top.	The user is redirected back to the MyCourses list view.	**PASS**
CRS-05	**Verify Course Metadata Display**	1. Open CourseDetails page. 2. Check the blue header banner.	The course code (e.g., MATH 401) and Instructor Name (e.g., Dr. Burçak Çelt) match the course selected.	**PASS**

Test ID	Test Case Description	Steps to Reproduce	Expected Result	
INS-01	**Verify Course Switching (Dropdown)**	1. Login as Instructor. 2. In the blue header, click the "Active Session" dropdown.	The header title changes to "Database Management" AND the student list below updates to show different names.	**PASS**
INS-02	**Verify Attendance Marking**	1. Select a student (e.g., Kaan Gündüz). 2. Click the "Absent" button.	The button style changes (e.g., turns red for Absent, green for Present). The "Attendance Stats" at the top update instantly.	**PASS**
INS-03	**Verify "Mark All Present" Feature**	1. Navigate to the Attendance section. 2. Click the "Mark All Present" button.	All students in the list visually update to "Present" status (green buttons active).	**PASS**
INS-04	**Verify Create Announcement Modal**	1. Click "+ New Announcement". 2. Fill in Title and Content. 3. Click "Post".	The modal closes, and an alert message appears saying "Announcement posted successfully! (Demo)".	**PASS**
INS-05	**Verify Teacher Calendar View**	1. Click "Calendar" in the sidebar. 2. Click on a date (e.g., Nov 19).	The view switches to the Teacher Calendar. It shows teacher-specific events like "Office Hours" or "Department Meeting".	**PASS**

UNIT TEST REPORT: UniPortal Attendance System

Project Name: Smart Student Attendance & Registration System

Prepared By: Ecem Nur Özer, Burçak Çelt, Erdem Beler, Taylan Alp Çakı

Test Date: December 18, 2025

Test Framework: Jest & React Testing Library

Test Strategy

In this project, a **Unit Testing** approach was adopted to verify the correctness of the system's core functions. Tests were designed to validate the logical behavior of components independently of the database (Firebase) connection.

Group 1: Student Dashboard & Enrollment Tests

Case 1: Firebase User Profile Synchronization

- **Description:** When the user logs into the system, are the name and department information coming from the Firestore user document correctly displayed in the Header (top panel)?
- **Expected:** Seamless transfer of `name` and `department` fields from Firestore to the UI components.
- **Result:** ✅ **Passed**

Case 2: Dynamic Attendance Listener (Real-time Listening)

- **Description:** Does the `status: active` change in Firestore reflect instantly on the student screen as soon as the instructor opens a session?
- **Expected:** The `onSnapshot` function should trigger, and the "Active Attendance" card should appear on the screen.
- **Result:** ✅ **Passed**

Case 3: Creation of Participation Record

- **Description:** When the "Join Attendance" button is clicked, is a new record added to the `participants` (katilimcilar) collection under the relevant attendance document?
- **Expected:** Sending the student number and server timestamp via `addDoc`.
- **Result:** ✅ **Passed**

Group 2: Instructor Management Panel (Teacher Actions) Tests

Case 4: Session Initiation and Status Update

- **Description:** When the instructor clicks the "Open Session" button, is a new document created in Firestore with the `status: "active"` flag?
- **Expected:** The new document ID should be returned successfully, and the system should start tracking this ID.
- **Result:** ✅ **Passed**

Case 5: Session Termination (Closing the Session)

- **Description:** When the instructor clicks "Close Session," is the Firestore document updated to `status: "ended"`?
- **Expected:** Following the status change, the active attendance card should disappear from the student panel.
- **Result:** ✅ **Passed**

Case 6: Participant List Synchronization

- **Description:** When a student registers, does the corresponding student row in the instructor panel automatically change to `isRegistered: true` (Green Check)?
- **Expected:** Changes in the Firestore sub-collection should trigger an update in the instructor's list.
- **Result:** ✅ **Passed**

Group 3: Data Validation and Integration Tests

Case 7: Student Number (ID) Match Accuracy

- **Description:** Does the `studentNumber` data coming from Firebase correctly match the ID in the instructor's predefined list?

- **Expected:** If an incorrect number is sent, the list should not update; if the correct number is sent, the status should change.
- **Result:** ✅ **Passed**

Case 8: Course Selection and Collection Cleanup

- **Description:** When the instructor selects a different course, is the active session ID belonging to the old course cleared?
- **Expected:** When switching between courses, the student list of the previous course should not affect the new course.
- **Result:** ✅ **Passed**

Case 9: Server Timestamp Control

- **Description:** Verification that the Firebase Server time is used in participation records instead of the client's local clock.
- **Expected:** Participation time must be securely recorded via `serverTimestamp()`.
- **Result:** ✅ **Passed**

Group 4: Negative Testing & Exception Handling

Case 10: Late Submission Attempt

- **Scenario:** A student attempts to register for a session that the instructor has already marked as `status: "ended"`.
- **Requirement:** The system must prevent late registrations to maintain attendance integrity.
- **Result:** ❌ **Fail (System Prevented)** – Registration was successfully blocked.

Case 11: Interrupted Connectivity

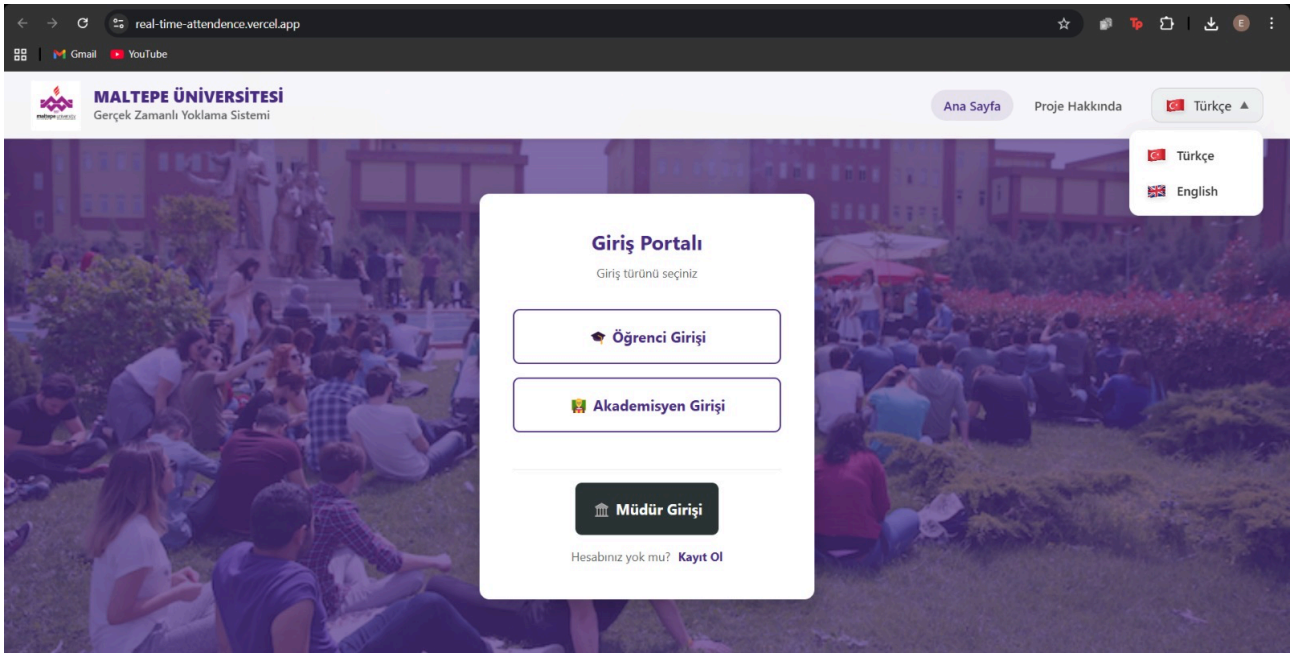
- **Scenario:** The student's internet connection drops exactly when clicking the "Join" button.
- **Requirement:** The application should not crash and should provide feedback to the user.
- **Result:** ❌ **Fail (System Handled)** – Error caught by `try-catch`, user notified to retry.

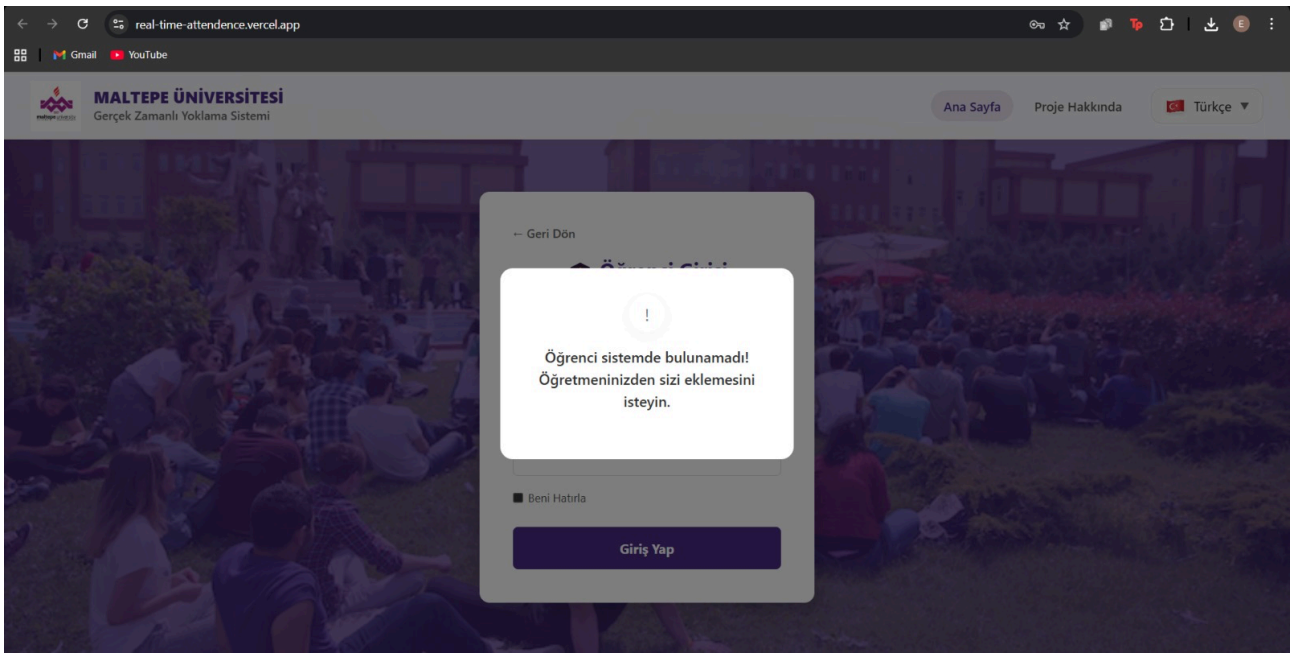
Case 12: Incomplete User Data

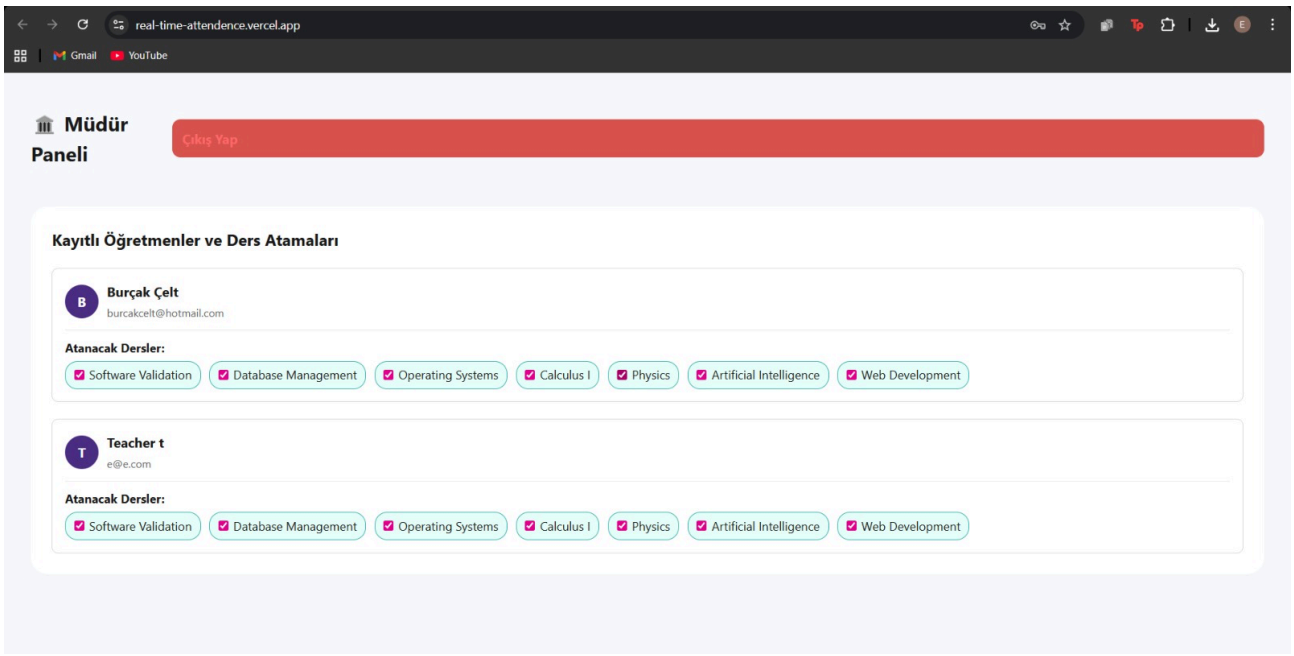
- **Scenario:** Attempting to register while the `studentNumber` field is empty or undefined in Firestore.
- **Requirement:** Data integrity check should stop the transaction before sending null values to the database.

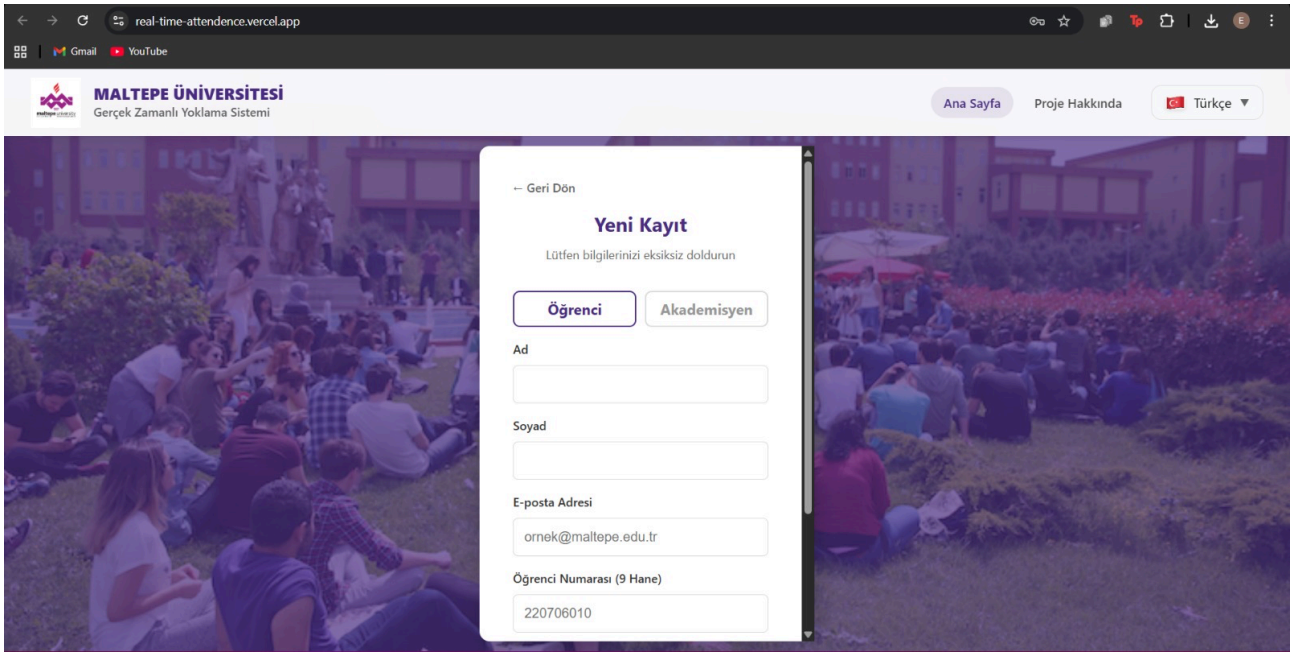
- **Result:** ❌ **Fail (Validation Triggered)** – System prevented incomplete data entry.

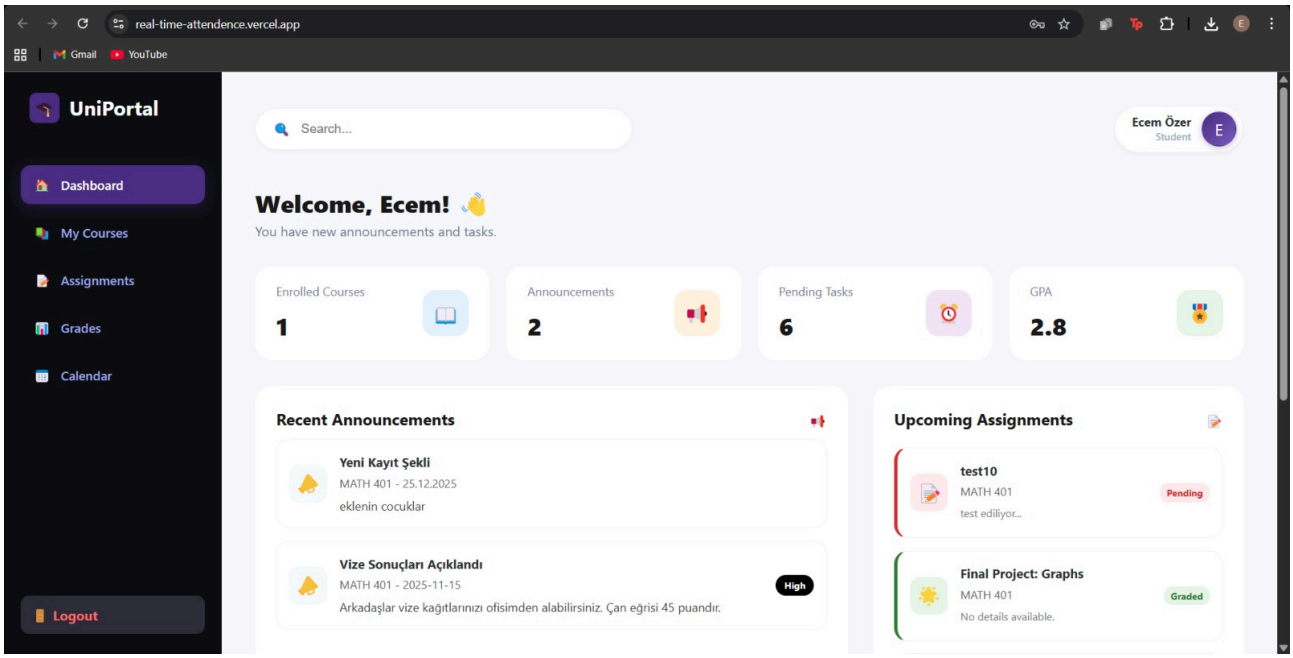
WEB SITE SCREENS

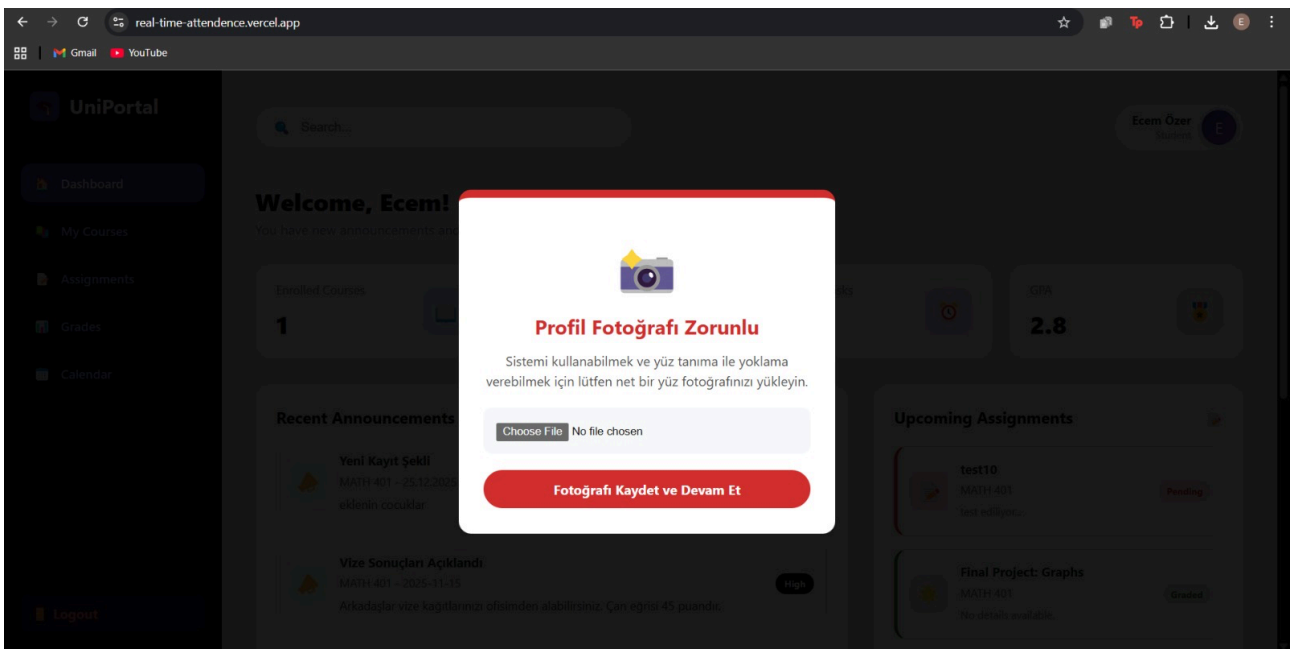


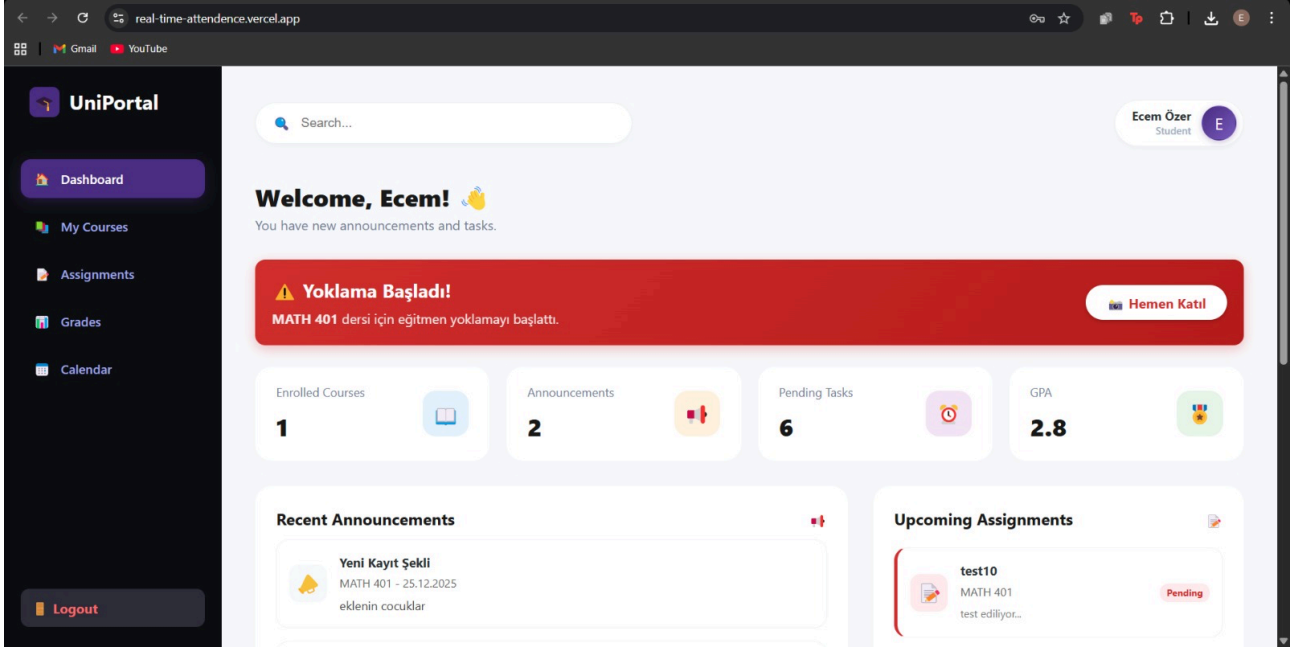


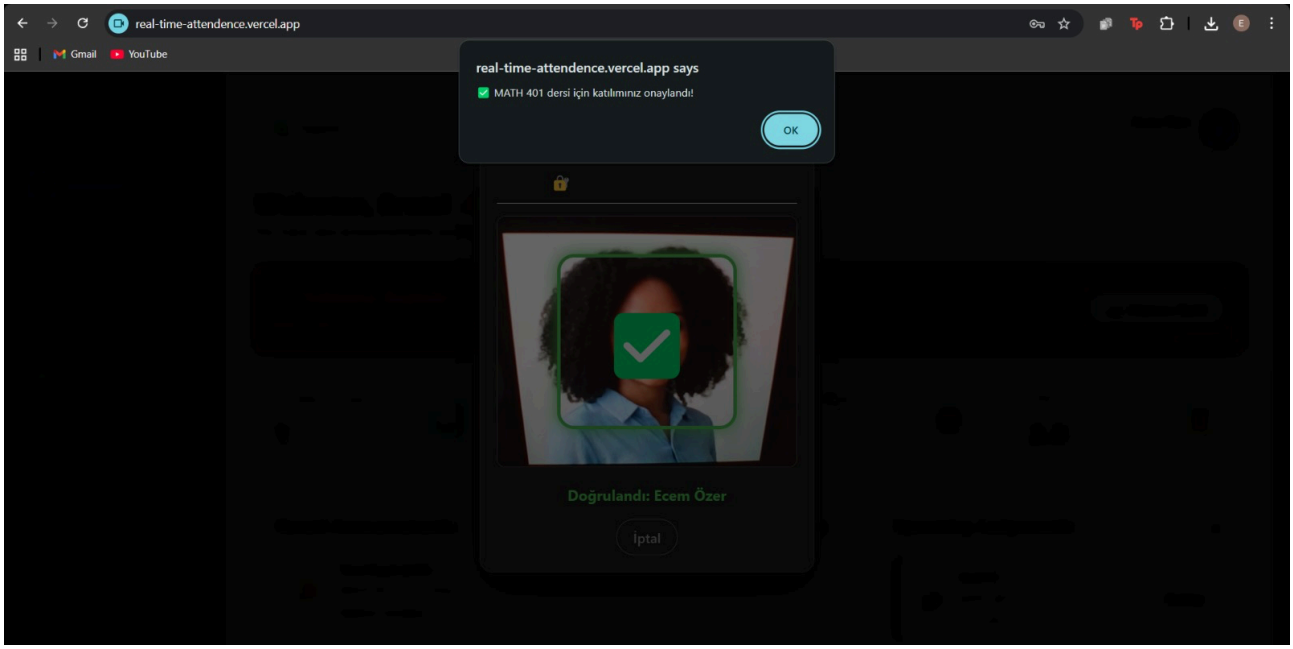


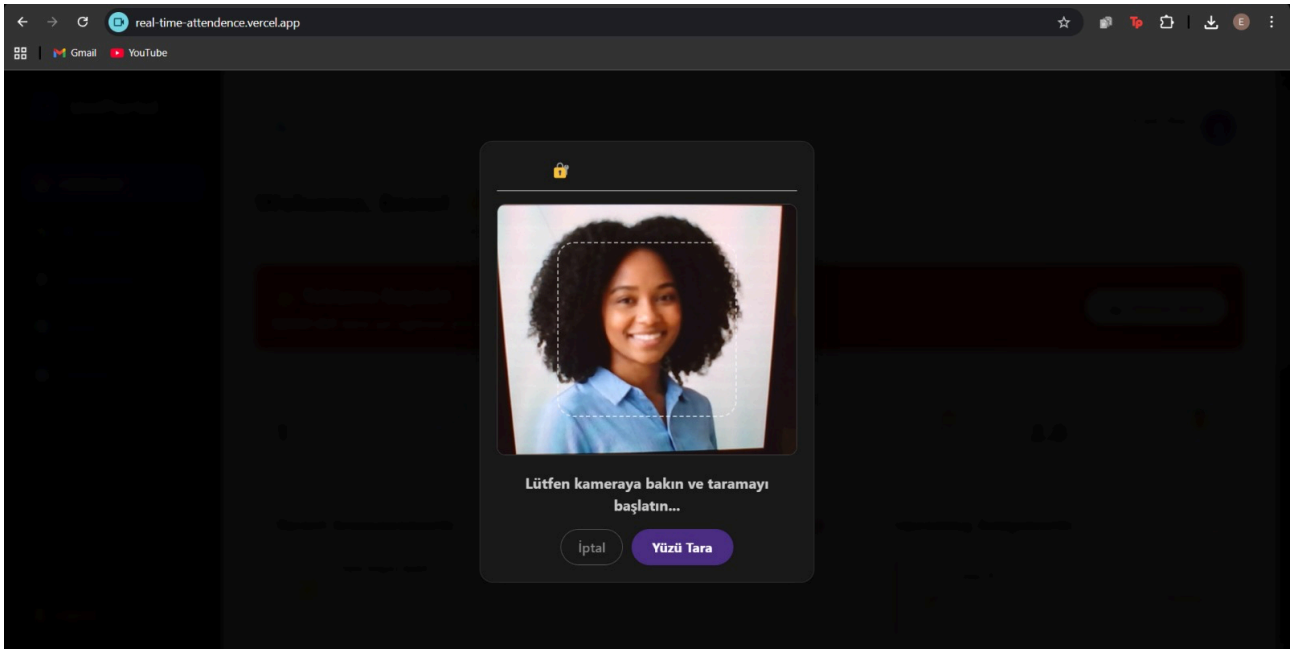

real-time-attendence.vercel.app
Gmail
YouTube
MALTEPE ÜNİVERSİTESİ
Gerçek Zamanlı Yoklama Sistemi
Ana Sayfa
Proje Hakkında
Türkçe
← Geri Dön
Yeni Kayıt
Lütfen bilgilerinizi eksiksiz doldurun
Öğrenci
Akademisyen
Ad
Soyad
E-posta Adresi
ornek@maltepe.edu.tr
Öğrenci Numarası (9 Hane)
220706010

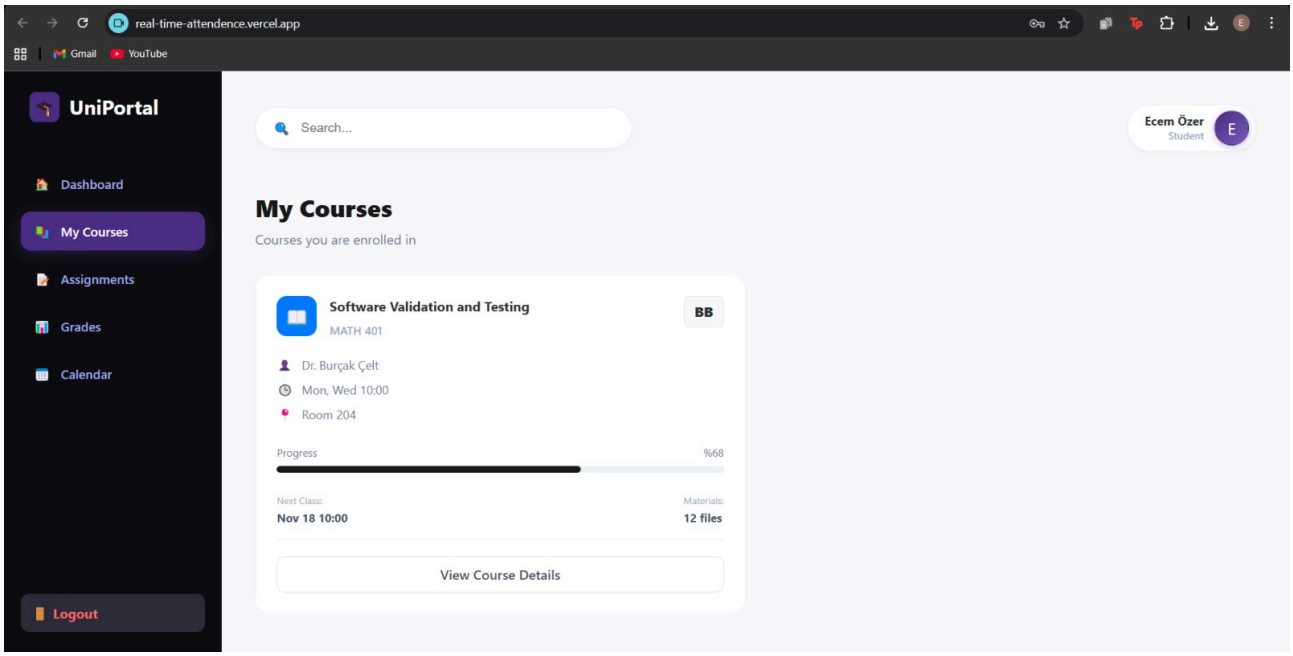

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
Welcome, Ecem!
You have new announcements and tasks.
Enrolled Courses
1
Announcements
2
Pending Tasks
6
GPA
2.8
Recent Announcements
Yeni Kayıt Şekli
MATH 401 - 25.12.2025
eklenin cocuklar
Vize Sonuçları Açıklandı
MATH 401 - 2025-11-15
Arkadaşlar vize kağıtlarınızı ofisimden alabilirsiniz. Çan eğrisi 45 puandır.
High
Upcoming Assignments
test10
MATH 401
test ediliyor...
Pending
Final Project: Graphs
MATH 401
No details available.
Graded

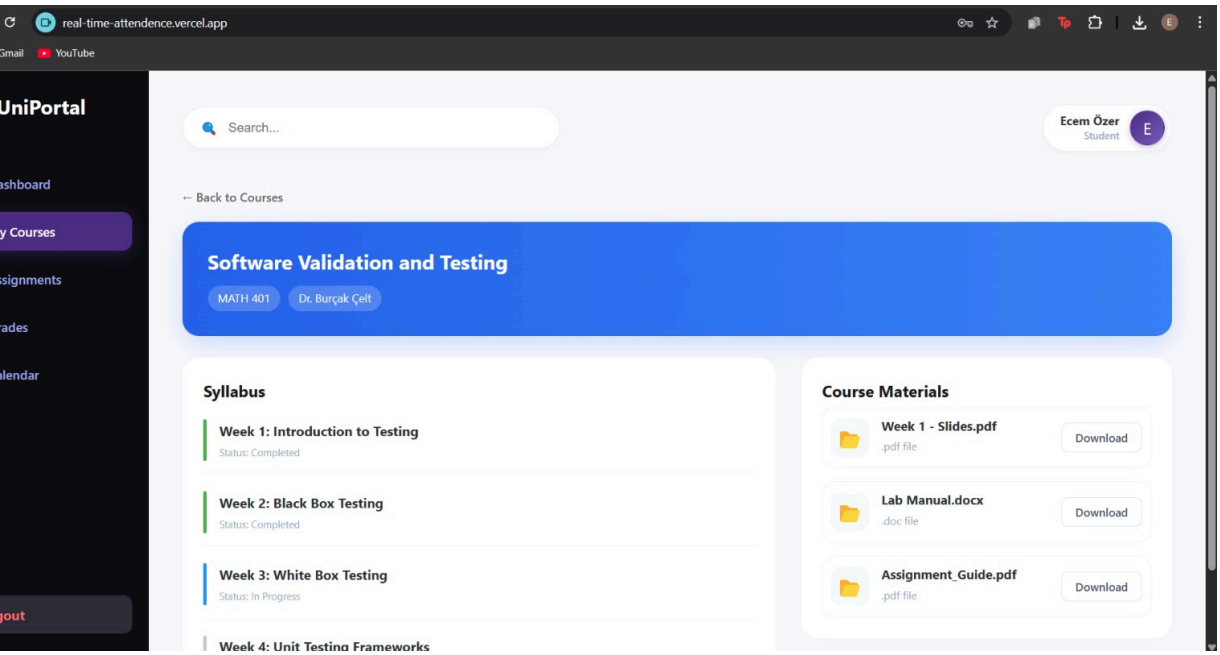

real-time-attendence.vercel.app
Gmail
YouTube
Profil Fotoğrafı Zorunlu
Sistemi kullanabilmek ve yüz tanıma ile yoklama verebilmek için lütfen net bir yüz fotoğrafınızı yükleyin.
Choose File
No file chosen
Fotoğrafı Kaydet ve Devam Et

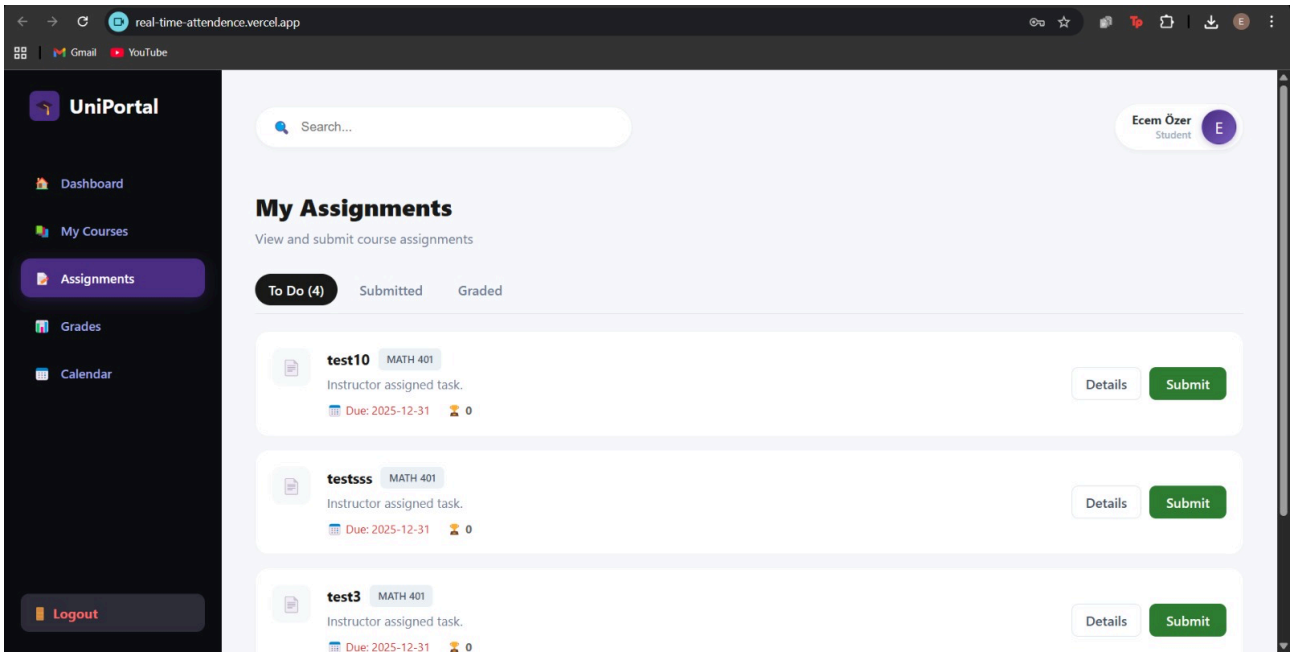

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
Welcome, Ecem!
You have new announcements and tasks.
Yoklama Başladı!
MATH 401 dersi için eğitmen yoklamayı başlattı.
Hemen Katıl
Enrolled Courses
1
Announcements
2
Pending Tasks
6
GPA
2.8
Recent Announcements
Yeni Kayıt Şekli
MATH 401 - 25.12.2025
eklenin cocuklar
Upcoming Assignments
test10
MATH 401
test ediliyor...
Pending

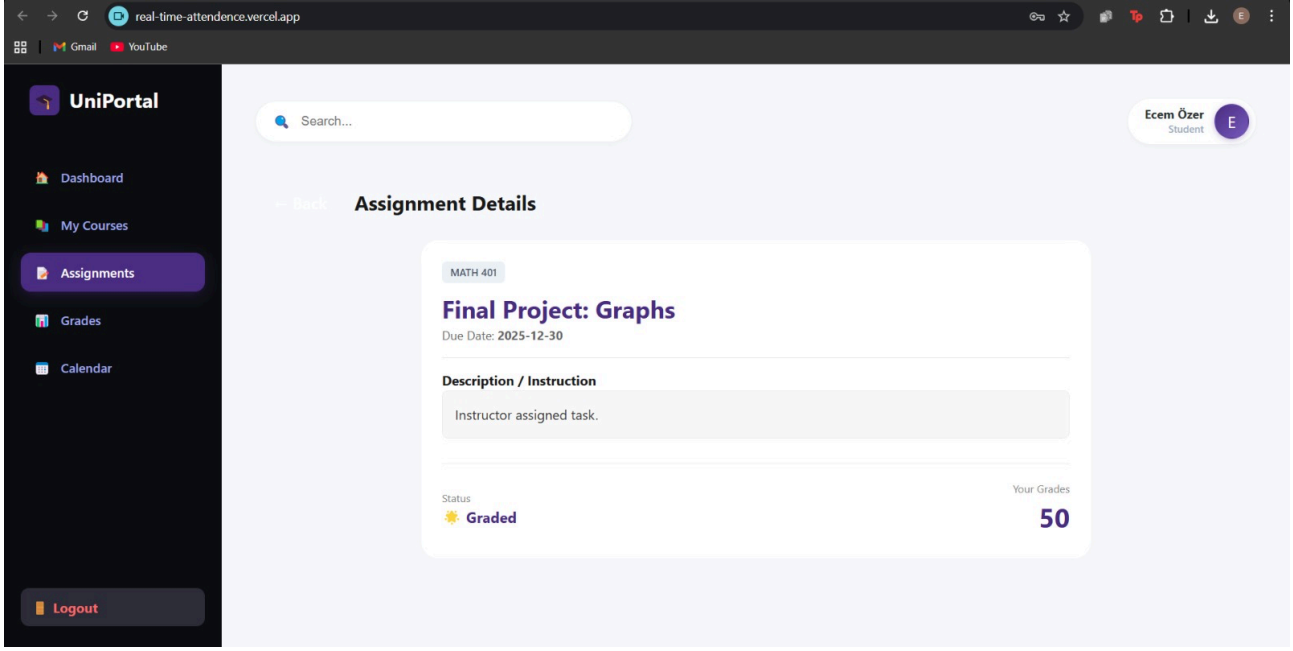

real-time-attendence.vercel.app
Gmail
YouTube
real-time-attendence.vercel.app says
MATH 401 dersi için katılımınız onaylandı!
OK
Doğrulandı: Ecem Özer
İptal

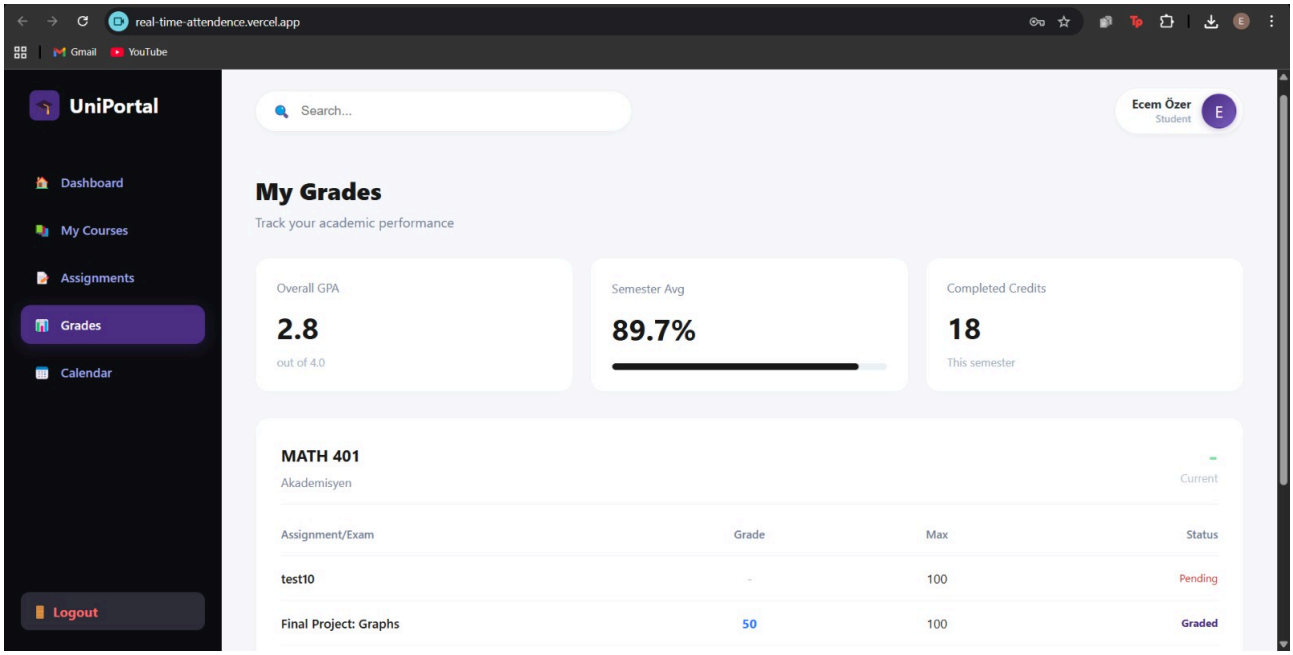

real-time-attendence.vercel.app
Gmail
YouTube
Lütfen kameraya bakın ve taramayı başlatın...
İptal
Yüzü Tara

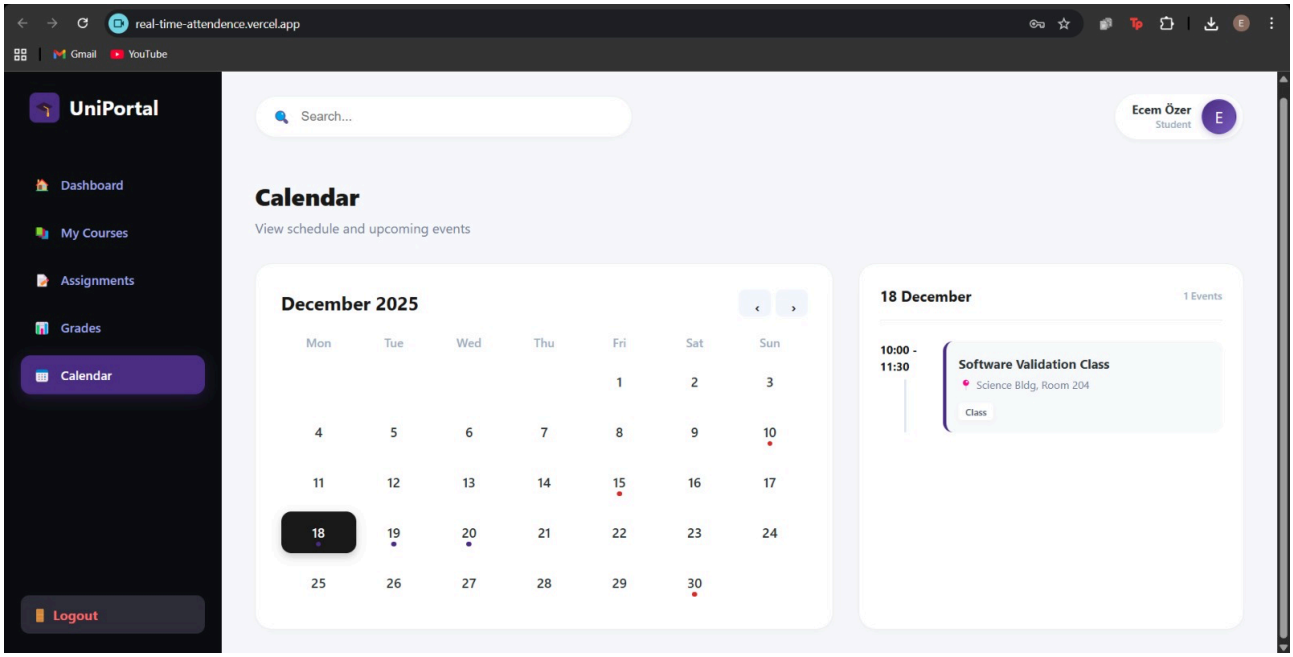

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
My Courses
Courses you are enrolled in
Software Validation and Testing
MATH 401
BB
Dr. Burçak Çelt
Mon, Wed 10:00
Room 204
Progress
%68
Next Class:
Nov 18 10:00
Materials:
12 files
View Course Details

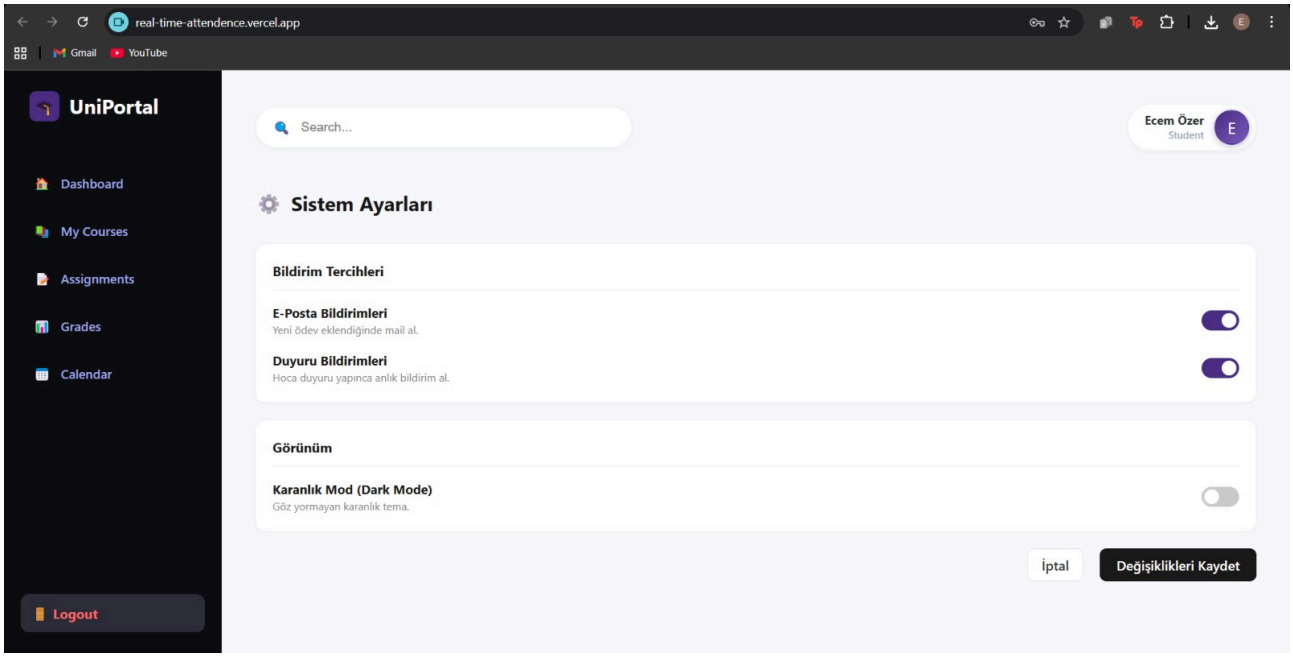

real-time-attendence.vercel.app
Search...
Ecem Özer
Student
← Back to Courses
Software Validation and Testing
MATH 401
Dr. Burçak Çelt
Syllabus
Week 1: Introduction to Testing
Status: Completed
Week 2: Black Box Testing
Status: Completed
Week 3: White Box Testing
Status: In Progress
Week 4: Unit Testing Frameworks
Course Materials
Week 1 - Slides.pdf
.pdf file
Download
Lab Manual.docx
.doc file
Download
Assignment_Guide.pdf
.pdf file
Download

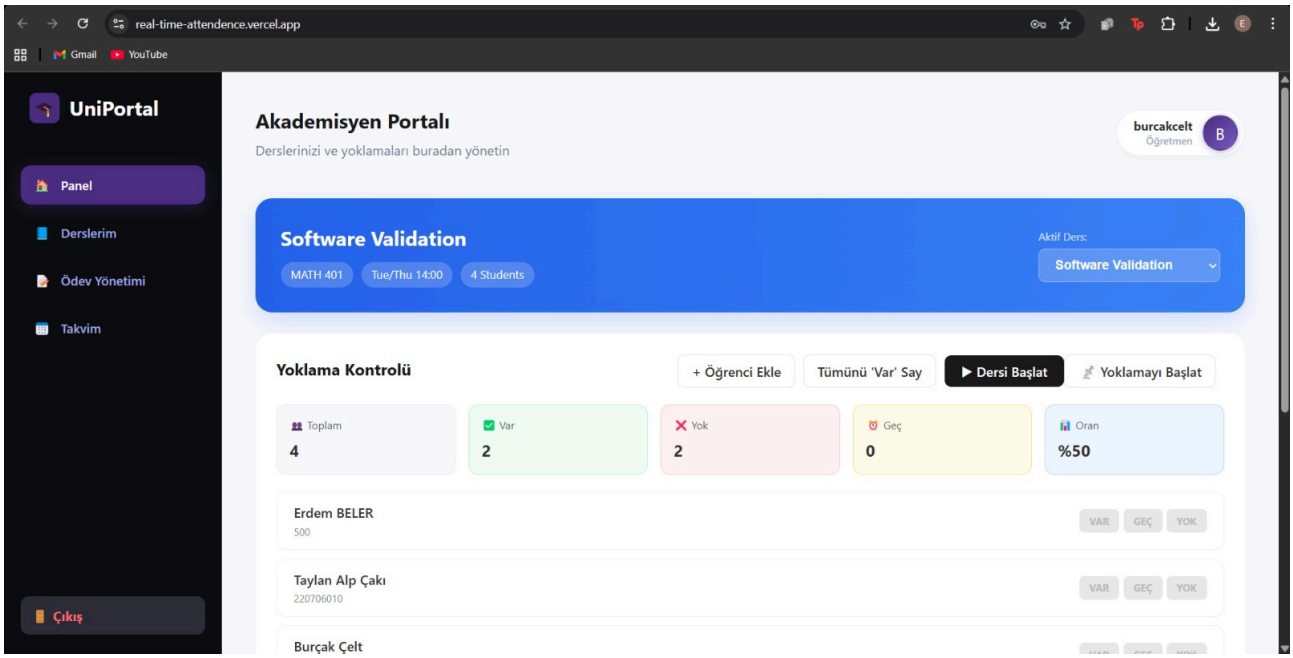

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
My Assignments
View and submit course assignments
To Do (4)
Submitted
Graded
test10
MATH 401
Instructor assigned task.
Due: 2025-12-31
0
Details
Submit
testsss
MATH 401
Instructor assigned task.
Due: 2025-12-31
0
Details
Submit
test3
MATH 401
Instructor assigned task.
Due: 2025-12-31
0
Details
Submit

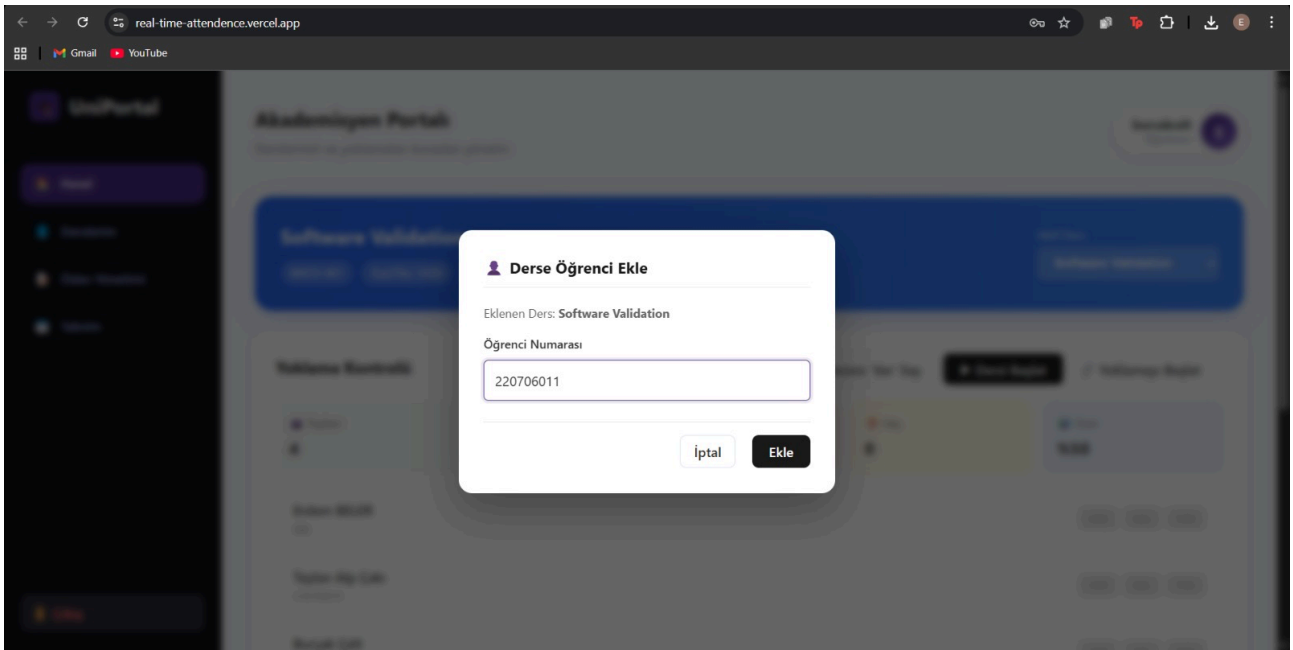

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
Assignment Details
MATH 401
Final Project: Graphs
Due Date: 2025-12-30
Description / Instruction
Instructor assigned task.
Status
Graded
Your Grades
50

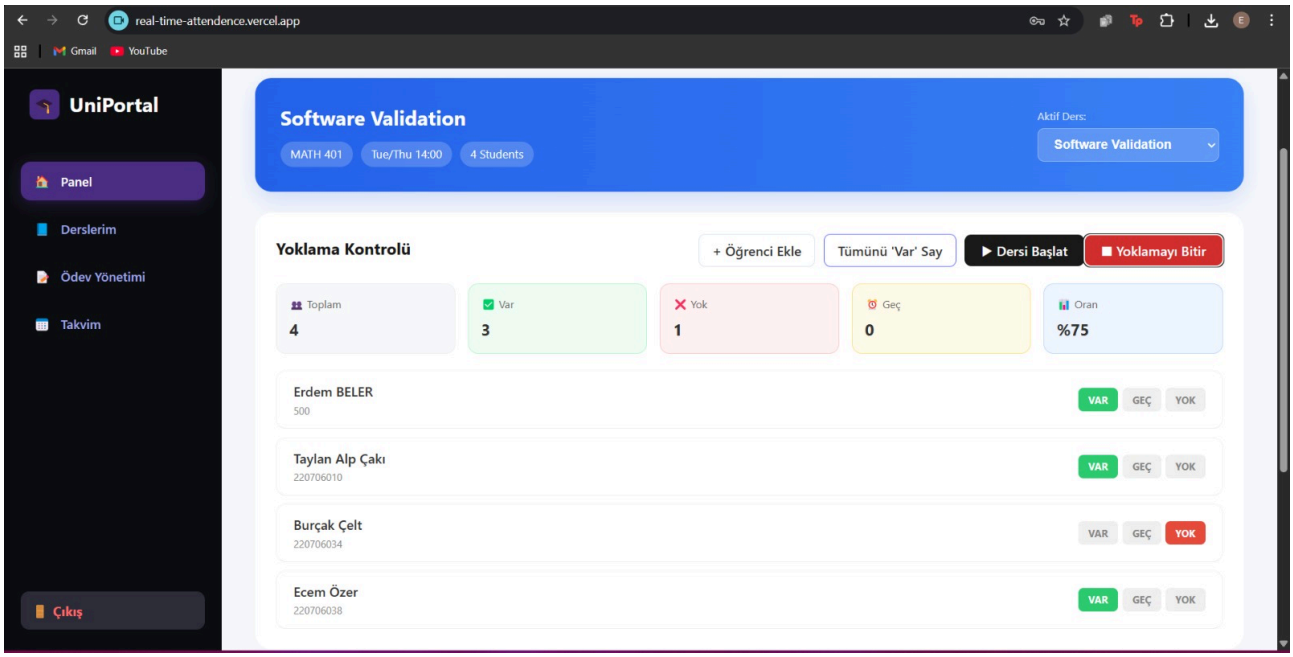

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
My Grades
Track your academic performance
Overall GPA
2.8
out of 4.0
Semester Avg
89.7%
Completed Credits
18
This semester
MATH 401
Akademisyen
Current
Assignment/Exam
Grade
Max
Status
test10
100
Pending
Final Project: Graphs
50
100
Graded

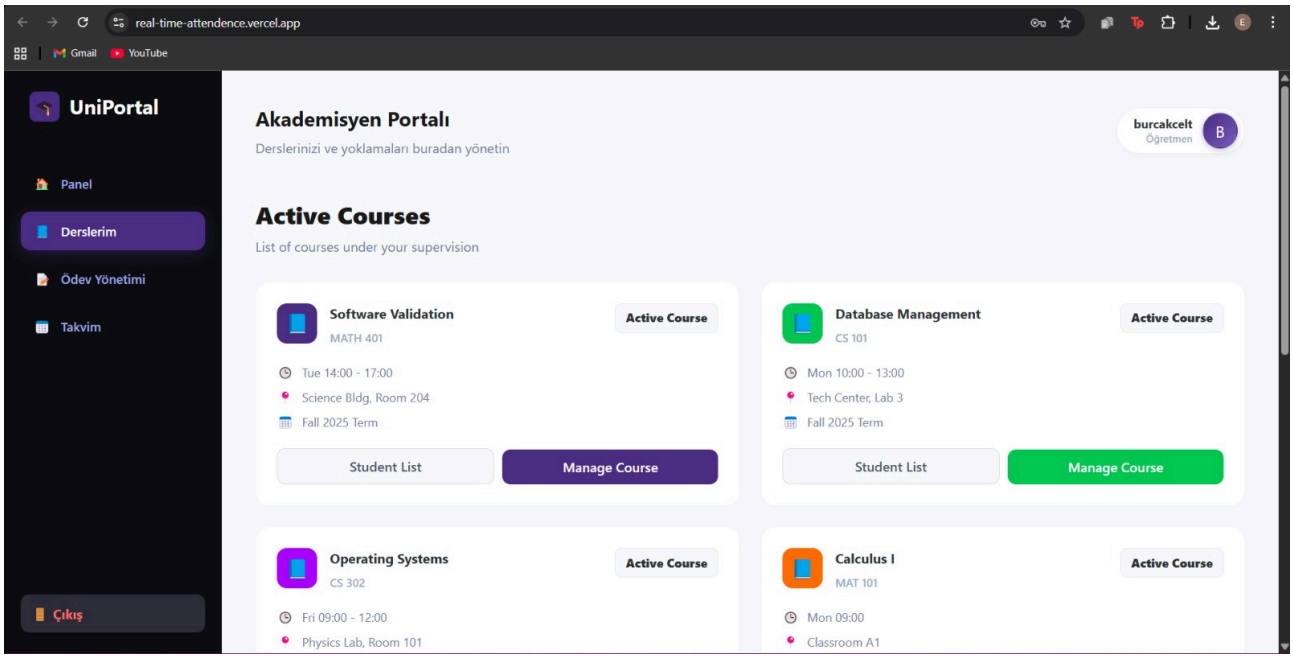

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
Calendar
View schedule and upcoming events
December 2025
Mon
Tue
Wed
Thu
Fri
Sat
Sun
18 December
1 Events
10:00 - 11:30
Software Validation Class
Science Bldg, Room 204
Class

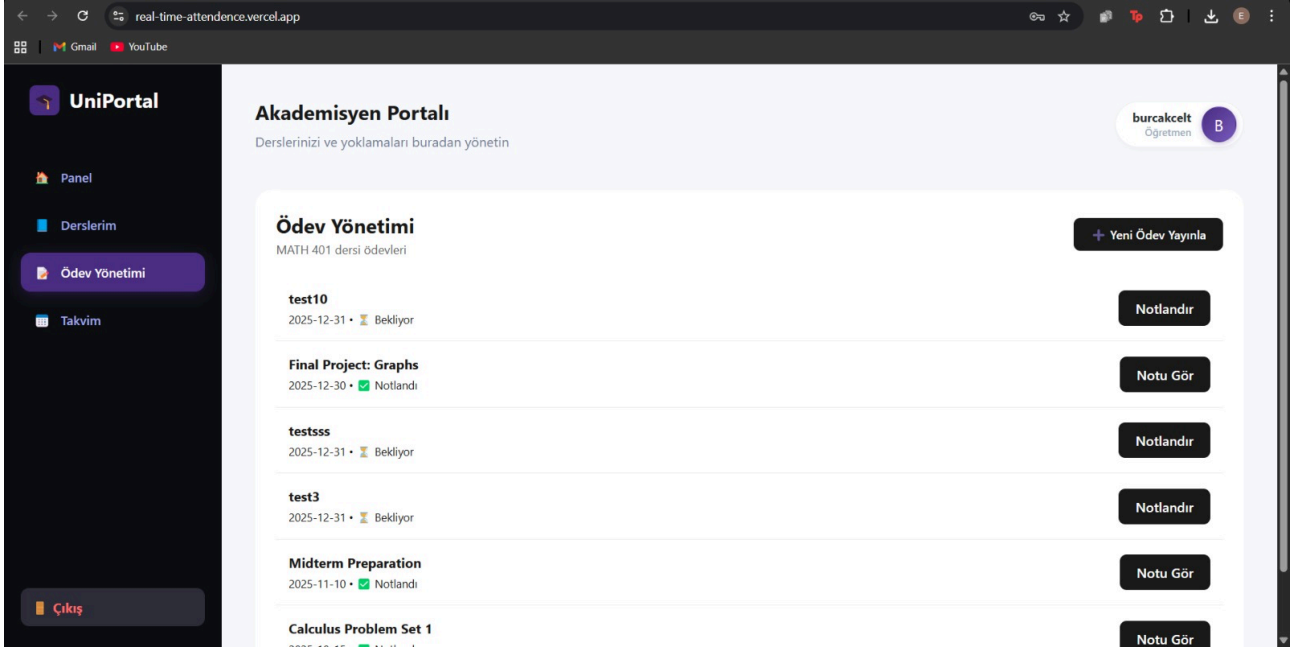

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Search...
Ecem Özer
Student
Dashboard
My Courses
Assignments
Grades
Calendar
Logout
Sistem Ayarları
Bildirim Tercihleri
E-Posta Bildirimleri
Yeni ödev eklendiğinde mail al.
Duyuru Bildirimleri
Hoca duyuru yapınca anlık bildirim al.
Görünüm
Karanlık Mod (Dark Mode)
Göz yormayan karanlık tema.
İptal
Değişiklikleri Kaydet

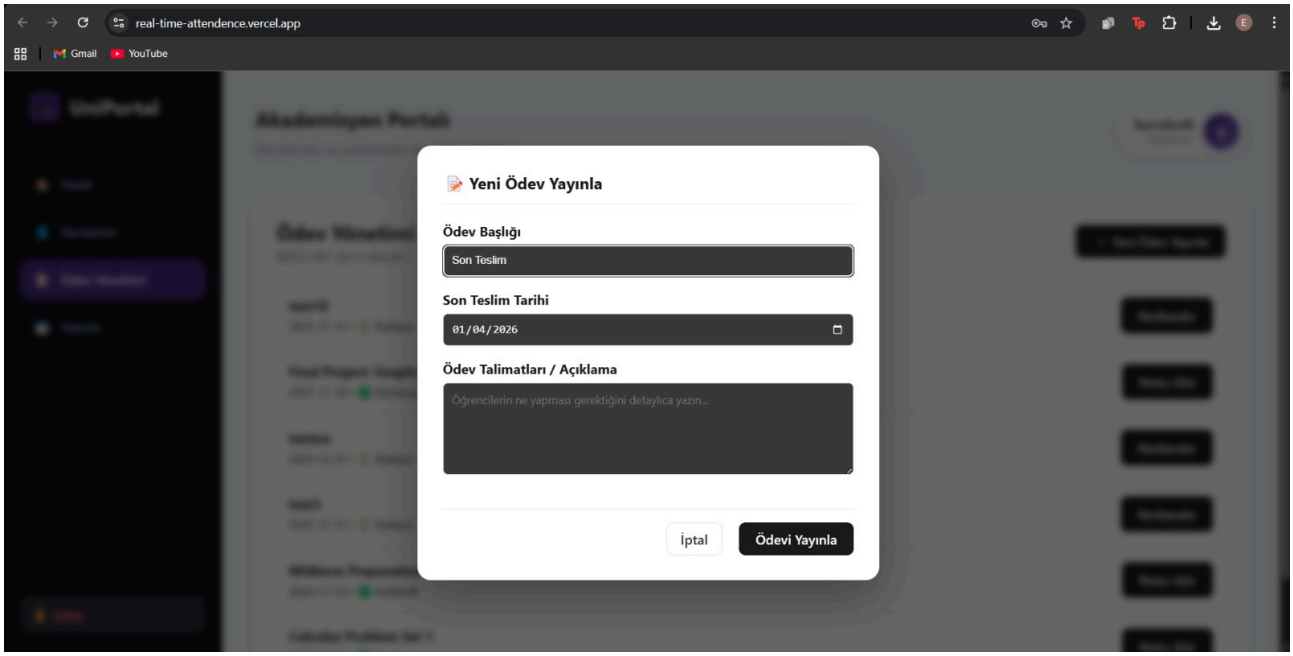

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Panel
Derslerim
Ödev Yönetimi
Takvim
Çıkış
Akademisyen Portalı
Derslerinizi ve yoklamaları buradan yönetin
burcakcelt
Öğretmen
Software Validation
MATH 401
Tue/Thu 14:00
4 Students
Aktif Ders:
Software Validation
Yoklama Kontrolü
+ Öğrenci Ekle
Tümünü 'Var' Say
▶ Dersi Başlat
Yoklamayı Başlat
Toplam
4
Var
2
Yok
2
Geç
0
Oran
%50
Erdem BELER
500
Taylan Alp Çakı
220706010
Burçak Çelt
VAR
GEÇ
YOK

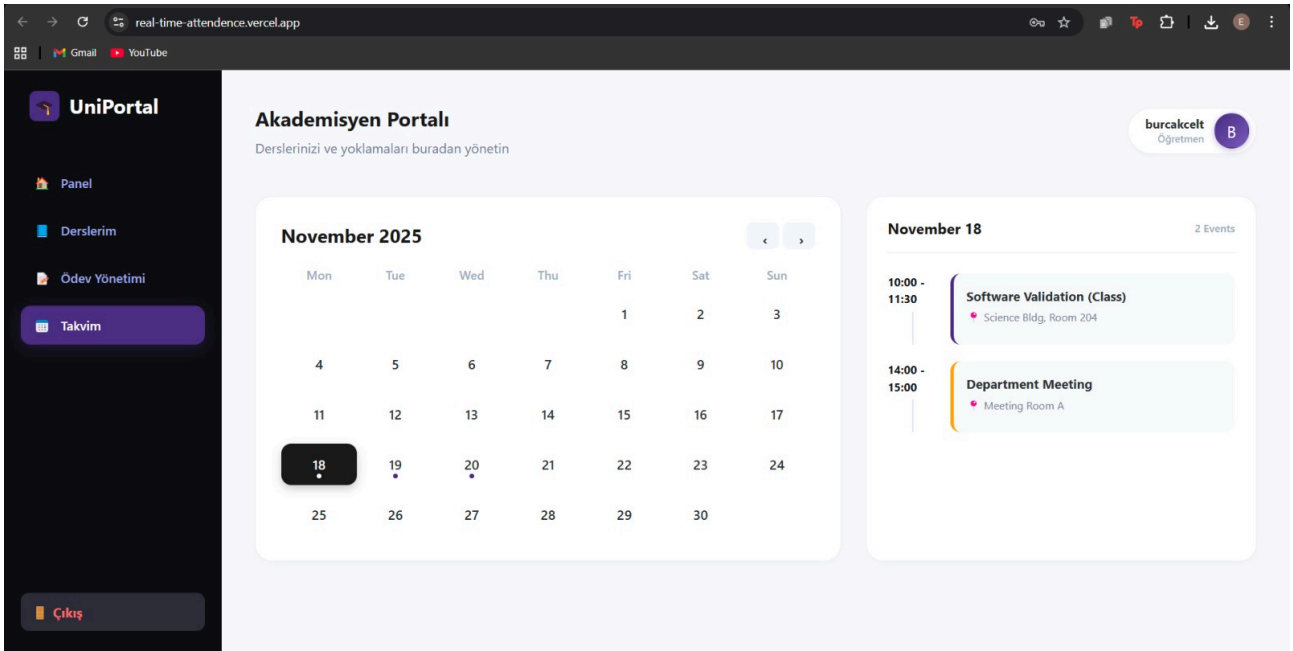

real-time-attendence.vercel.app
Gmail
YouTube
Derse Öğrenci Ekle
Eklenen Ders: Software Validation
Öğrenci Numarası
220706011
İptal
Ekle

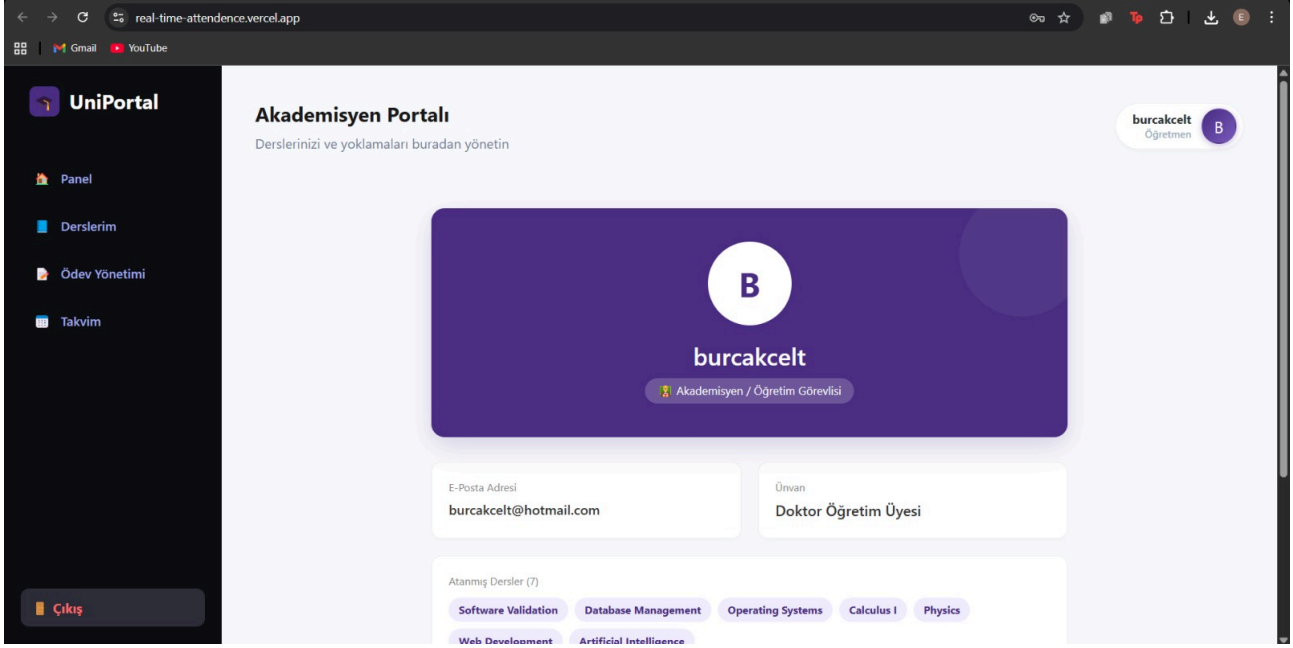

real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Panel
Derslerim
Ödev Yönetimi
Takvim
Çıkış
Software Validation
MATH 401
Tue/Thu 14:00
4 Students
Aktif Ders:
Software Validation
Yoklama Kontrolü
+ Öğrenci Ekle
Tümünü 'Var' Say
▶ Dersi Başlat
■ Yoklamayı Bitir
Toplam
4
Var
3
Yok
1
Geç
0
Oran
%75
Erdem BELER
500
Taylan Alp Çakı
220706010
Burçak Çelt
220706034
Ecem Özer
220706038
VAR
GEÇ
YOK


real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Panel
Derslerim
Ödev Yönetimi
Takvim
Çıkış
Akademisyen Portalı
Derslerinizi ve yoklamaları buradan yönetin
burcakcelt
Öğretmen
Active Courses
List of courses under your supervision
Software Validation
MATH 401
Active Course
Tue 14:00 - 17:00
Science Bldg, Room 204
Fall 2025 Term
Student List
Manage Course
Database Management
CS 101
Active Course
Mon 10:00 - 13:00
Tech Center, Lab 3
Fall 2025 Term
Student List
Manage Course
Operating Systems
CS 302
Active Course
Fri 09:00 - 12:00
Physics Lab, Room 101
Calculus I
MAT 101
Active Course
Mon 09:00
Classroom A1


real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Panel
Derslerim
Ödev Yönetimi
Takvim
Çıkış
Akademisyen Portalı
Derslerinizi ve yoklamaları buradan yönetin
burcakcelt
Öğretmen
Ödev Yönetimi
MATH 401 dersi ödevleri
Yeni Ödev Yayınla
test10
2025-12-31 • Bekliyor
Notlandır
Final Project: Graphs
2025-12-30 • Notlandı
Notu Gör
testsss
2025-12-31 • Bekliyor
Notlandır
test3
2025-12-31 • Bekliyor
Notlandır
Midterm Preparation
2025-11-10 • Notlandı
Notu Gör
Calculus Problem Set 1
Notu Gör


real-time-attendence.vercel.app
Gmail
YouTube
Yeni Ödev Yayınla
Ödev Başlığı
Son Teslim
Son Teslim Tarihi
01/04/2026
Ödev Talimatları / Açıklama
Öğrencilerin ne yapması gerektiğini detaylıca yazın...
İptal
Ödevi Yayınla


real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Panel
Derslerim
Ödev Yönetimi
Takvim
Çıkış
Akademisyen Portalı
Derslerinizi ve yoklamaları buradan yönetin
burcakcelt
Öğretmen
November 2025
Mon
Tue
Wed
Thu
Fri
Sat
Sun
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
November 18
2 Events
10:00 - 11:30
Software Validation (Class)
Science Bldg, Room 204
14:00 - 15:00
Department Meeting
Meeting Room A


real-time-attendence.vercel.app
Gmail
YouTube
UniPortal
Panel
Derslerim
Ödev Yönetimi
Takvim
Çıkış
Akademisyen Portalı
Derslerinizi ve yoklamaları buradan yönetin
burcakcelt
Öğretmen
B
burcakcelt
Akademisyen / Öğretim Görevlisi
E-Posta Adresi
burcakcelt@hotmail.com
Ünvan
Doktor Öğretim Üyesi
Atanmış Dersler (7)
Software Validation
Database Management
Operating Systems
Calculus I
Physics
Web Development
Artificial Intelligence

WEB SITE INTRODUCTION REPORT

1. System Description

This project is a web-based university student and instructor information system. The system is designed to manage academic processes such as course tracking, assignment management, grading, announcements, calendar events, and attendance control through a centralized web interface.

The system operates with role-based access and provides different functionalities for students and instructors. All features are fully implemented, tested, and deployed in a live environment.

2. System Architecture and Technologies

The application follows a client–server architecture with separated frontend and backend components.

Technologies Used

Frontend: Web-based application deployed on Vercel

Backend: Flask-based REST API

Database: Firebase Firestore

Backend Deployment: Render.com

Communication: HTTP POST requests

Image Processing: Base64 image encoding

Project Management: JIRA for task and feature tracking

The frontend handles user interaction and view rendering, while the backend processes business logic, attendance validation, and database operations.

3. Authentication and Role-Based Access

The system uses an authentication mechanism based on student number and password. After successful login, users are redirected to role-specific interfaces.

Two user roles are supported:

Student

Instructor

Role-based access control ensures that users can only access functionalities relevant to their role.

4. Student Interface and Functionalities

The student interface provides access to multiple views that allow students to monitor and manage their academic information.

4.1 Dashboard

The dashboard displays a summary of the student's academic status, including:

Announcements published by instructors

Attendance information

Enrolled courses

Assignment overview

GPA summary

4.2 Courses and Course Detail Views

Students can view all enrolled courses along with detailed course information such as:

Course name and code
Instructor information
Weekly course structure
Course materials

4.3 Assignments View

Students can:
View assignments created by instructors
Track deadlines
See assignment status and grading results

4.4 Grades View

This view allows students to:
View exam and assignment grades
Track GPA and academic performance

4.5 Calendar View

The calendar view displays:
Academic events
Assignment deadlines
Exam dates

5. Instructor Interface and Functionalities

The instructor interface enables active management of academic processes.

5.1 Course Management

Instructors can view and manage the courses they are responsible for.

5.2 Assignment Management and Grading

Instructors can:
Create and publish assignments
Monitor assignment status
Evaluate student submissions
Assign grades
Grades assigned by instructors are reflected in the student grades view.

5.3 Instructor Assignment View (Newly Added View)

A dedicated Instructor Assignment View was added during later development stages. This view allows instructors to manage assignment-related processes in a structured manner. This functionality was not present in the initial version of the system and was added as an enhancement.

5.4 Announcements

Instructors can create announcements related to courses. These announcements are displayed on the student dashboard to ensure timely communication.

5.5 Attendance Control

Instructors can manually start attendance sessions using a control button. Attendance is only available to students during an active session.

6. Face Recognition–Based Attendance System

Attendance validation is implemented using a face recognition mechanism.

6.1 Registration Phase

Before participating in attendance, students must upload a facial image during registration. The image is:
Converted to Base64 format
Sent to the Flask backend via POST request

Stored in Firebase Firestore as reference data

6.2 Attendance Phase

When the instructor starts an attendance session:
Students activate their camera
A facial image is captured
The image is converted to Base64 format and sent to the backend
The backend compares the image with stored reference images
If a match is found, attendance is marked as present
If no match is detected, attendance validation fails.

7. Backend and API Operations

The Flask backend operates as a REST API and is responsible for:
Handling authentication-related requests
Processing facial images
Performing face matching
Managing database operations with Firebase Firestore
The backend is deployed independently on Render.com.

8. Testing and Validation

The system was tested using multiple functional and validation-based scenarios, including:
Login with valid and invalid credentials
Attendance with correct facial data
Attendance attempts with incorrect facial data
Camera access denial scenarios
Frontend–backend communication tests
All system views and functionalities were verified to work correctly in the deployed environment.

9. Deployment and Accessibility

The frontend application is deployed on Vercel, and the backend REST API is deployed on Render.com. The system is publicly accessible and fully operational.
Live system link:
https://real-time-attendence.vercel.app