



Bilkent University

Department of Computer Engineering

CS 353 Database Systems

Project Final Report

Online Course Platform

Group 1

Abdullah Can Alpay - 21702686 - Section 2

Cemre Biltekin - 21802369 - Section 1

Mustafa Yaşar - 21702808 - Section 1

Oğuz Kaan İmamoğlu - 21702233 - Section 1

Instructor: Uğur Güdükbay

Teaching Assistant: Mustafa Can Çavdar

Application System Description	3
Final Entity-Relationship (E/R) Diagram	4
Final List of Tables	5
person	5
student	5
course_creator	5
course	5
enrolls	5
adds_to_cart	6
adds_to_wishlist	6
progresses	6
admin	6
lecture	7
assignment	7
submitted_assignments	7
question	7
announcement	7
discount	8
refund	8
refund_requests	8
complaint	8
feedback	8
certificate	9
student_complaints	9
student_feedbacks	9
earns	9
note	10
asks	10
answers	10
Implementation Details	11
Contribution of Team Members	11
Advanced Database Features	12
Reports	12
Views	13
Triggers	14
Constraints	14
Secondary Indices	15
User's Manual	15
Getting Started - Login & Sign Up	15

Student User Manual	17
Course Creator User Manual	24
Admin User Manual	26
Website	29

1.Application System Description

The project is a web-based database application which is an online course platform that provides courses for students and enables course creation for instructors. There are three types of end users. The first one is the students. Students are able to search for courses, add courses to their wishlists and carts, buy the courses that they desire, add funds to their wallets. They can keep track of their progress on different courses, once they complete a course, they can give feedback and ratings to that course and receive a certificate of completion. Also, they are able to request a refund for a bought course, with valid reasons. They can reach recent information from their notifications. Another end user in the platform is the course creator. Course creators can publish courses, add lectures and assignments to these courses and make announcements for these courses. They can keep track of how much they earn via their own wallets. Students and course creators can participate in Q&A sessions where students ask questions and the course creator answers. The last type of users is admin. Admin can offer discounts for courses, and it is up to the course creator of that course to decide whether to accept the discount or not. Site admins can see complaints from students and should accept or decline refund requests of students. Admin can see statistics (report) about the application.

A course consists of several lectures and may contain assignments. Each of these lectures and assignments must be completed by students if they want to get a completion certificate for those courses. Students can take notes for lectures and associate these notes with lectures. Courses have question & answer sections. In these sections, students can ask their questions related to courses and the creator answers these questions.

2. Final Entity-Relationship (E/R) Diagram

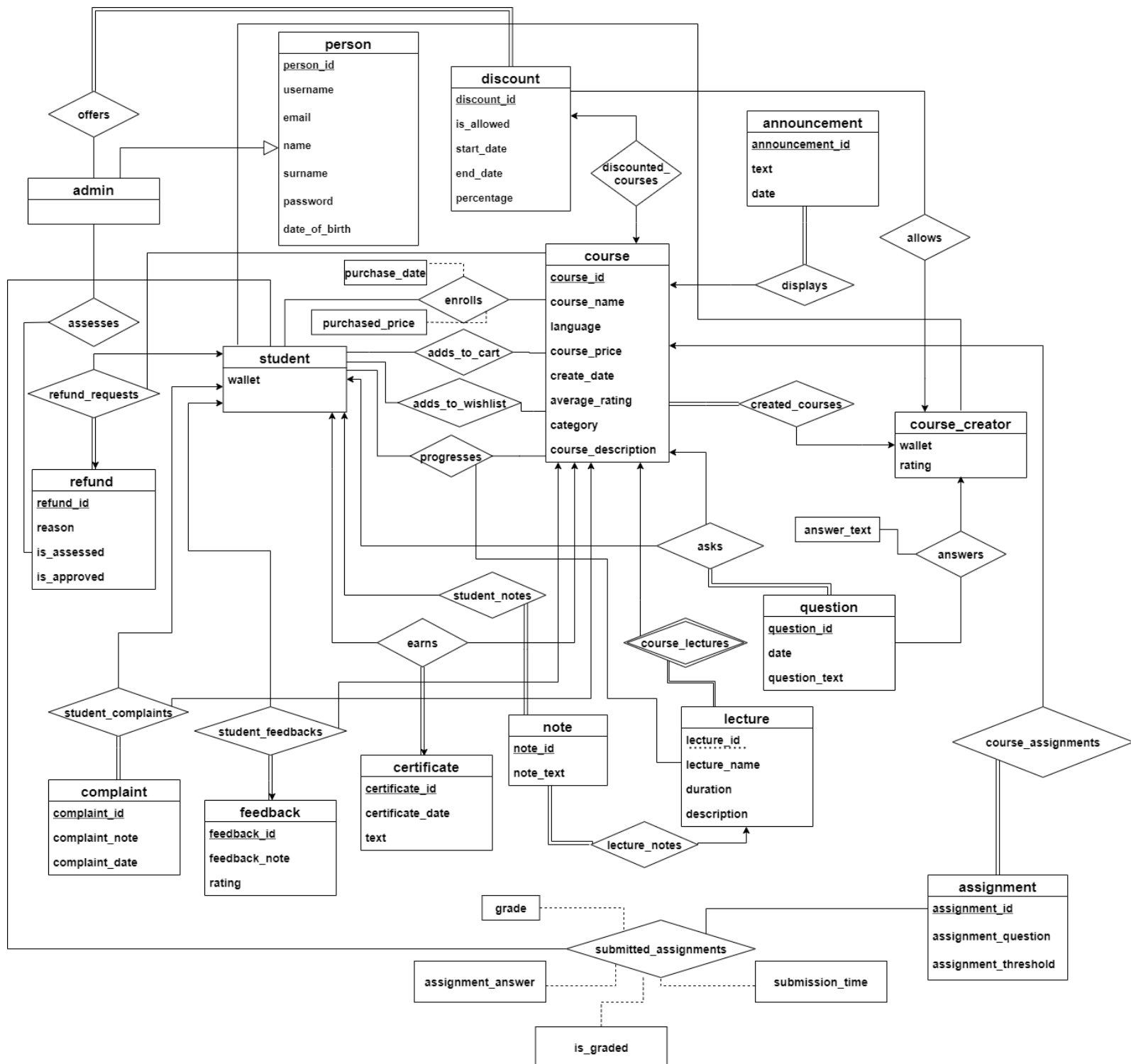


Figure 1: Final E/R Diagram of Online Course Platform

3. Final List of Tables

person

Relational Schema: person(person_id, username, email, name, surname, password, date_of_birth)

Candidate Keys: {person_id}, {username}, {email}

Primary Key: {person_id}

Foreign Key: None

student

Relational Schema: student(student_id, wallet)

Candidate Key: {student_id}

Primary Key: {student_id}

Foreign Key: student_id references person(person_id)

course_creator

Relational Schema: course_creator(course_creator_id, wallet, rating)

Candidate Key: {course_creator_id}

Primary Key: {course_creator_id}

Foreign Key: course_creator_id references person(person_id)

course

Relational Schema: course(course_id, course_name, language, course_price, create_date, average_rating, category, course_description, course_creator_id)

Candidate Key: {course_id}

Primary Key: {course_id}

Foreign Key: course_creator_id references course_creator(course_creator_id)

enrolls

Relational Schema: enrolls(student_id, course_id, purchased_price, purchase_date)

Candidate Key: {student_id, course_id}

Primary Key: {student_id, course_id}

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)

adds_to_cart

Relational Schema: adds_to_cart(student_id, course_id)

Candidate Key: {student_id, course_id}

Primary Key: {student_id, course_id}

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)

adds_to_wishlist

Relational Schema: adds_to_wishlist(student_id, course_id)

Candidate Key: {student_id, course_id}

Primary Key: {student_id, course_id}

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)

progresses

Relational Schema: progresses(student_id, course_id, lecture_id)

Candidate Key: {student_id, course_id, lecture_id}

Primary Key: {student_id, course_id, lecture_id}

Foreign Keys:

- student_id references student(student_id)
- {course_id, lecture_id} references lecture(lecture_id)
- {course_id} references course(course_id)

admin

Relational Schema: admin(admin_id)

Candidate Key: {admin_id}

Primary Key: {admin_id}

Foreign Key: admin_id references person(person_id)

lecture

Relational Schema: lecture(course_id, lecture_id, lecture_name, duration, description)

Candidate Key: {course_id, lecture_id}

Primary Key: {course_id, lecture_id}

Foreign Key: course_id references course(course_id)

assignment

Relational Schema: assignment(assignment_id, assignment_question, assignment_threshold, course_id)

Candidate Key: {assignment_id}

Primary Key: {assignment_id}

Foreign Key: course_id references course(course_id)

submitted_assignments

Relational Schema: submitted_assignments(assignment_id, student_id, submission_time, assignment_answer, grade, is_graded)

Candidate Key: {assignment_id, student_id, submission_time}

Primary Key: {assignment_id, student_id, submission_time}

Foreign Keys:

- assignment_id references assignment(assignment_id)
- student_id references student(student_id)

question

Relational Schema: question(question_id, date, question_text)

Candidate Keys: {question_id}

Primary Keys: {question_id}

Foreign Keys: None

announcement

Relational Schema: announcement(announcement_id, text, date, course_id)

Candidate Key: {announcement_id}

Primary Key: {announcement_id}

Foreign Key: course_id references course(course_id)

discount

Relational Schema: discount(discount_id, is_allowed, start_date, end_date, percentage, discounted_course_id, alloweer_course_creator_id)

Candidate Keys: {discount_id}, {discounted_course_id}

Primary Key: {discount_id}

Foreign Key:

- discounted_course_id references course(course_id)
- alloweer_course_creator_id references course_creator(course_creator_id)

refund

Relational Schema: refund(refund_id, reason, is_assessed, is_approved)

Candidate Keys: {refund_id}

Primary Key: {refund_id}

Foreign Key: None

refund_requests

Relational Schema: refund_requests(refund_id, student_id, course_id)

Candidate Keys: {refund_id}, {student_id, course_id}

Primary Key: {refund_id}

Foreign Keys:

- refund_id references refund(refund_id)
- student_id references student(student_id)
- course_id references course(course_id)

complaint

Relational Schema: complaint(complaint_id, complaint_note, complaint_date)

Candidate Key: {complaint_id}

Primary Key: {complaint_id}

Foreign Key: None

feedback

Relational Schema: feedback(feedback_id, feedback_note, rating)

Candidate Key: {feedback_id}

Primary Key: {feedback_id}

Foreign Key: None

certificate

Relation Schema: certificate(certificate_id, date, text)

Candidate Key: {certificate_id}

Primary Key: {certificate_id}

Foreign Key: None

student_complaints

Relational Schema: student_complaints(complaint_id, student_id, course_id,)

Candidate Key: {complaint_id}

Primary Key: {complaint_id}

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)
- complaint_id references complaint(complaint_id)

student_feedbacks

Relational Schema: student_feedbacks(feedback_id, student_id, course_id)

Candidate Keys: { feedback_id }, { student_id, course_id }

Primary Key: { student_id, course_id }

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)
- feedback_id references feedback(feedback_id)

earns

Relational Schema: earns(student_id, course_id, certificate_id)

Candidate Keys: {student_id, course_id}, {certificate_id}

Primary Key: { student_id, course_id }

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)
- certificate_id references certificate(certificate_id)

note

Relational Schema: note(note_id, student_id, lecture_id, course_id, note_text)

Candidate Key: {note_id}

Primary Key: {note_id}

Foreign Keys:

- student_id from student(student_id)
- (lecture_id, course_id) from lecture(lecture_id, course_id)

asks

Relational Schema: asks(question_id, student_id, course_id)

Candidate Key: {question_id}

Primary Keys: {question_id}

Foreign Keys:

- student_id references student(student_id)
- course_id references course(course_id)
- question_id references question(question_id)

answers

Relational Schema: answers(question_id, course_creator_id, answer_text)

Candidate Key: {question_id}

Primary Key: {question_id}

Foreign Keys:

- question_id references question(question_id)
- course_creator_id references course_creator(course_creator_id)

4. Implementation Details

MySQL was used for the database management system. PHP was used to develop the back-end and the front-end of the web application. Java was used to create the database. Javascript, CSS, HTML5 were used to develop the front-end (UI) of the web application. Bootstrap was used as a framework (CSS, HTML5 and JS). AJAX was used.

One challenge in implementation was to fetch appropriate data with a button click to display appropriate modal content to the user (i.e. in the implementation of accepting/rejecting refund requests as admin) and have the user update data with submission via modal. Hidden inputs were implemented in the form component that was submitted by the button that opens up the modal to carry information to the modal. Additionally a Javascript function (onClick function) had to be written to make modal content change with different fetched data by the button. A problem in assignment functionality occurred during implementation due to missing an attribute in our design. The assignment status was vague, so `is_graded` attribute was added to the `submitted_assignment` to indicate a clearer grading status. It enabled the student to see its assignment status as successful, unsuccessful or not graded which is user friendly and allows the student to make meaningful attempts by looking at the assignment status first.

Contribution of Team Members

Mustafa Yaşar

Course market page and filters in it, cart and wishlist page, course page from student's and course creator's perspective and course page redirected from market, making complaints by student and reading them by admin, getting certificate, sign up pages and operations, log in operations, publishing new courses have been implemented.

Cemre Biltekin

All assignment (additional) functionalities & pages of users, management and display pages of refund requests & discount offers by admin, admin page, and log out have been implemented.

Oğuz Kaan İmamoğlu

Course related functionalities for students (see enrolled courses and announcements, watch lectures and take notes), notification related features and statistics/reports page of site have been implemented.

Abdullah Can Alpay

All question & answer related functionalities, all feedback related functionalities and myCertificates page implemented. Additionally, add a discount page for course creator where course creator can accept/reject the offer.

5. Advanced Database Features

Triggers, constraints and secondary indices SQL statements can be found in our Java code.

Reports

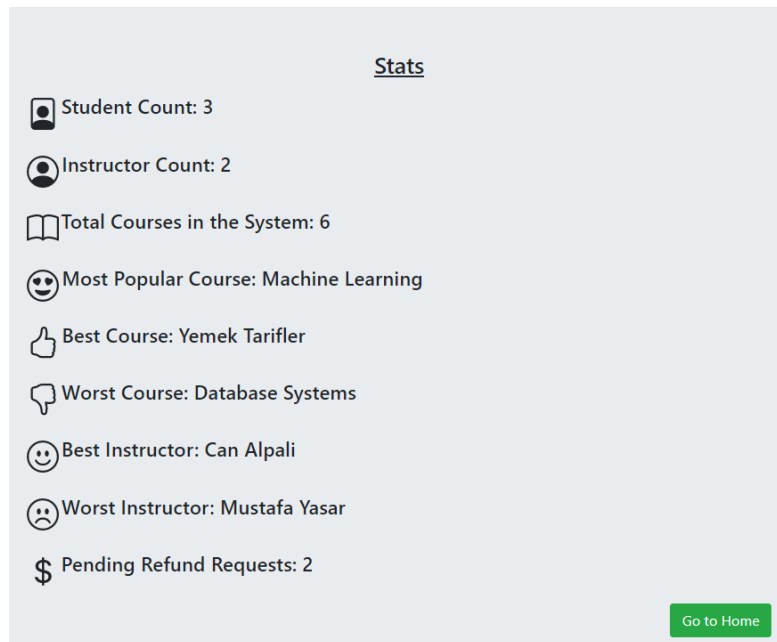


Figure 2: Stats Page Seen by Admin with Reports Queries

```
SELECT      name, surname
FROM        person
WHERE       person_id = (SELECT course_creator_id FROM course_creator WHERE
RATING = (SELECT MAX(RATING) FROM course_creator LIMIT 1) LIMIT 1)
SELECT      course_id, COUNT(course_id) AS occurrence
FROM        enrolls
GROUP BY    course_id ORDER BY occurrence DESC LIMIT 1
```

These reports find the instructor with the highest rating, in other words the best instructor, and the course that has the highest number of students, in other words the most popular course with aggregate functions.

```
SELECT      C.course_id, C.course_name, C.language, C.average_rating, C.category,
            P.name, P.surname, D.percentage, D.start_date, D.end_date,
            D.is_allowed, (CASE WHEN CURRENT_DATE <= D.end_date AND
CURRENT_DATE >= D.start_date AND D.is_allowed THEN C.course_price *
((100 - D.percentage) / 100) ELSE C.course_price END) as price
FROM        course C LEFT OUTER JOIN discount D ON C.course_id =
D.discounted_course_id LEFT JOIN person P ON C.course_creator_id = P.person_id
```

With this complex query we can filter the course market, according to course's price, name, discount, category, language; course creator's name. Additionally, we can sort them by price (high to low, low to high) and rating (high to low) of the course.

[Apply Filter Options](#)

Course Name	Price	Language	Rating	Category	Instructor Name	Discount
Operating Systems	47.00	English	4.0	Information Technology	Ibrahim Korpeoglu	0
Introduction to Algorithms and Data Structures	90.00	English	3.3	Programming Languages	Ibrahim Korpeoglu	0
Digital Design	27.50	English	0.0	Information Technology	Can Aydin	50%
Computer Organisation	15.00	German	0.0	Programming Languages	Can Aydin	0
Murat ile Mutfaga Giris	100.00	Turkish	0.0	Gastronomy	Murat Korkmaz	50%
Turkiyede Ne Yenir	100.00	Turkish	0.0	Gastronomy	Murat Korkmaz	0

[Refresh](#)

Figure 3: Before applying the filter

[Apply Filter Options](#)

Course Name	Price	Language	Rating	Category	Instructor Name	Discount
Operating Systems	47.00	English	4.0	Information Technology	Ibrahim Korpeoglu	0
Digital Design	27.50	English	0.0	Information Technology	Can Aydin	50

[Refresh](#)

Figure 4: After Applying min price = 0, max price = 100, category = information technology

Views

Home	Refund Requests	Discounts	Stats	Complaints	Logout
<div> <p>Student: Mustafa Yasar - Course: Operating Systems</p> <p>2021-05-16</p> <p>This course is not OS course</p> </div> <div> <p>Student: Mustafa Yasar - Course: Database Management Systems</p> <p>2021-05-16</p> <p>This course is not good.</p> </div>					

Figure 5: Complaints Page Seen by Admin with Complaint_View

SQL View Statement for Complaints

```
CREATE VIEW complaint_view AS  
SELECT C.complaint_id, complaint_note, complaint_date, P.person_id, Co.course_id,  
name, surname, course_name  
FROM complaint C, student_complaints SC, person P, course Co  
WHERE  
C.complaint_id=SC.complaint_id AND P.person_id=SC.student_id AND  
SC.course_id=Co.course_id
```

This view stores the information about complaints. The information consists of the name of the student who made the complaint, the name of the course the complaint is made about, and the date the complaint was made. This view was used in the Complaints page on admin's side for ease of use and ease of seeing complaints made by students on particular courses.

Triggers

Trigger for Updating Average Rating of Course After Feedback Insertion

A trigger was implemented for after the insertion of a student feedback to update the average rating of the course. After a student gives feedback to a course, the course's rating should be calculated and updated again with the new feedback.

Trigger for Updating Average Rating of Course After Feedback Update

A trigger was implemented for after the update of a student feedback to update the average rating of a course. After a student changes his/her feedback for the course, the course's rating should be calculated and updated again with the updated feedback.

Trigger for Updating Rating for Course Creator After Feedback Update

The course creator's rating is determined by taking the average of his/her course's ratings. A trigger was implemented to calculate/recalculate the course creator's rating after a student makes an update on his/her feedback about a course offered by the course creator.

Trigger for Updating Rating for Course Creator After Feedback Insertion

The course creator's rating is determined by taking the average of his/her course's ratings. A trigger was implemented to calculate/recalculate the course creator's rating after a student sends feedback on one of his/her courses.

Constraints

A constraint for keeping wallet of student and course creator nonnegative, for keeping rating of course and course creator nonnegative, for checking valid category for course, for checking nonnegativeness of course price, for checking lecture duration being positive number and smaller than 1.5 hours, for checking assignment threshold and assignment grade being nonnegative and smaller than 100, for checking discount percentage to be between 0 and 100 have been implemented.

Secondary Indices

Person Table's Username Index

The search is made more efficient in the login process to search for the instance of a matching username by creating an index on *username* in the person table.

Announcement Table's Course ID Index

The search is made more efficient in the course page to search for the instance of its announcements by creating an index on *course_id* in the announcement table. This was implemented on a foreign key in this table for efficiency.

Course Table's Course Creator ID Index

The search is made more efficient to search for the instance of a course's course creator by creating an index on *course_creator_id* in the course table. This was implemented on a foreign key in this table for efficiency.

6. User's Manual

Getting Started - Login & Sign Up

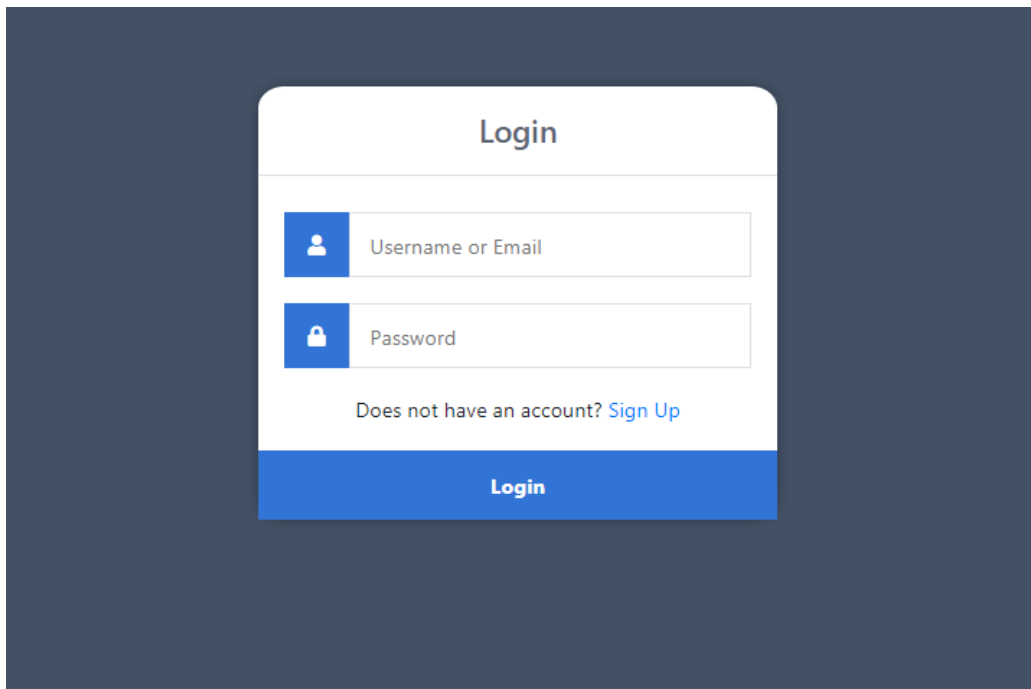


Figure 6: Login Page

To be able to use the features of the online course platform, you have to have a registered account first. On the login page if you already have an account, you are asked to enter your credentials (username/email and password) to have access to the application.

Sign Up to Coursemy

As a Student

Register as a Course Creator

Date of Birth:

Register Now

Already have an account? [Login](#)

Figure 7: Sign Up Page

If you do not have an account, there is a link on the login page “Does not have an account?” that directs you to the sign up page for registering as a new user. You can either sign up as a course creator or as a student. Administrator accounts are issued/authorized by the system. You have to fill the sign up form as specified by the system. In case an erroneous input is given, the system reports why the sign up has failed with appropriate error messages (i.e. username character constraints, uniqueness of email and username). In case you want to switch to the login page, you can click “Already have an account?” link.

After filling the sign up form successfully, pressing the Register Now button will create your account in the system and direct you to the login page. You can login to your account following the login procedures as explained above.

In case you want to log out, you can log out of your account using the Log Out link on the top navigation bar. This is available on every page.

Student User Manual

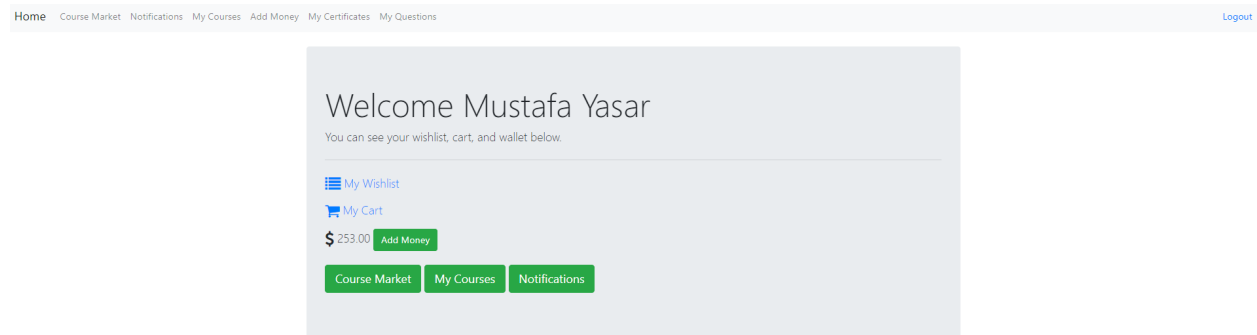


Figure 8: Student Welcome Page

When you log in as a student, the student home page welcomes you and lists the courses you have in the table if you have any. You can see your wishlist and cart by clicking “My Wishlist” or “My Cart”. You can add money to your wallet by clicking the “Add Money” button and specifying the amount you want to add. The amount can be at most 250\$.

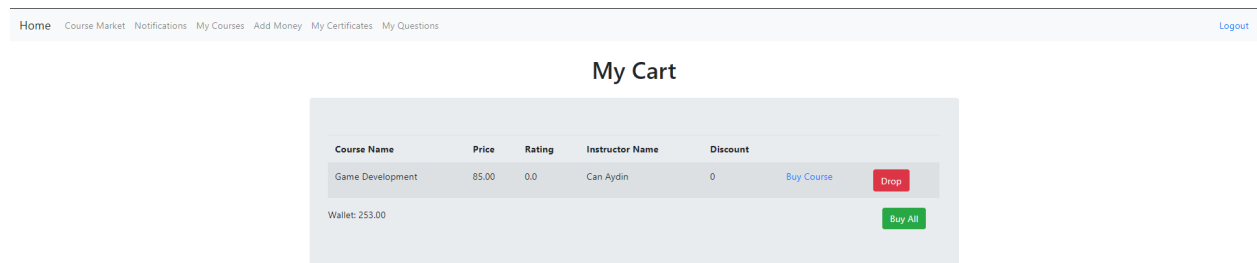


Figure 9: Student Cart Page

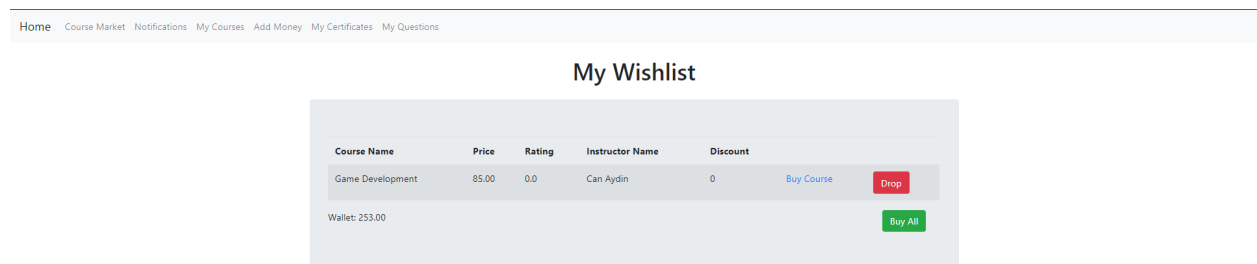


Figure 10: Student Wishlist Page

You can buy a course in your cart or in your wishlist by clicking “buy course”, or delete the course by clicking the “drop” button. You can click the “buy all” button to buy every course in your wishlist or cart.

[Home](#)
[Course Market](#)
[Notifications](#)
[My Courses](#)
[Add Money](#)
[My Certificates](#)
[My Questions](#)
[Logout](#)

List of Available Courses

Language
Select Language of the Course

Sort By
Sort By

Category
Select the Category of the Course

Apply Filter Options

Course Name	Price	Language	Rating	Category	Instructor Name	Discount
Operating Systems	47.00	English	2.5	Information Technology	Ibrahim Korpeoglu	0
Introduction to Algorithms and Data Structures	90.00	English	3.3	Programming Languages	Ibrahim Korpeoglu	0
Digital Design	27.50	English	0.0	Information Technology	Can Aydin	50%
Computer Organisation	15.00	German	0.0	Programming Languages	Can Aydin	0
Murat ile Mutfaga Giris	100.00	Turkish	0.0	Gastronomy	Murat Korkmaz	50%
Turkiyede Ne Yenir	100.00	Turkish	0.0	Gastronomy	Murat Korkmaz	0
Game Development	85.00	English	0.0	Game Development	Can Aydin	0
Ornek Course	60.00	German	0.0	Maths	Murat Korkmaz	40%

Refresh

Figure 11: Course Market

In order to see the course market, you can click the “Course Market” button. This button directs you to the course market where you can find every course with their information. You can specify as many filters as you wish while searching for a course. If you have applied filters, by clicking the “Refresh” button, you can list every course again.

[Home](#)
[Course Market](#)
[Notifications](#)
[My Courses](#)
[Add Money](#)
[My Certificates](#)
[My Questions](#)
[Logout](#)

Operating Systems

You have already enrolled in this course.

Course Instructor Ibrahim Korpeoglu

Language English

Course Description Operating System Concepts

Average Rating 2.5

Price 47.00\$

Discount There is no discount on this course.

You have already enrolled in this course.

Comments

Name Surname	Comment	Rate/5
Oguz Keser		4.0
Mustafa Yasar		1.0

Figure 12: Course Page from Market

When you click on a course in the course table, you are directed to the course page where you can see every information about that course. In this page, you can add the course to your cart and wishlist and you can see comments. If you have already purchased it, you cannot add it to your cart or your wishlist.

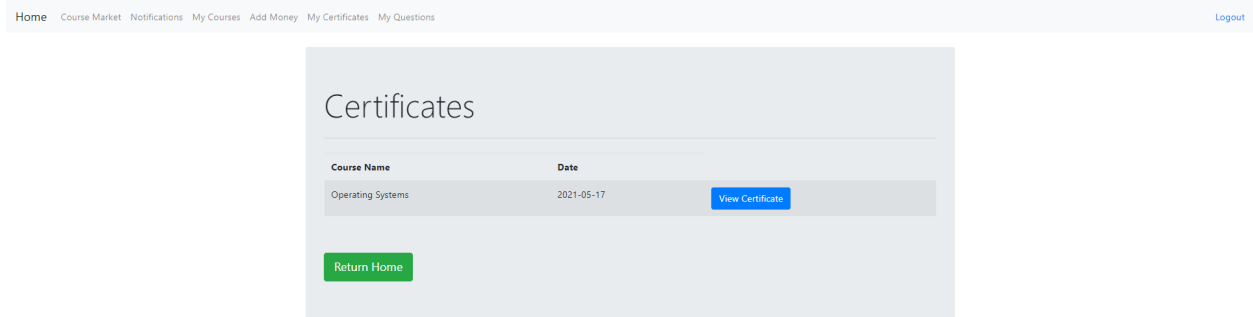


Figure 13: My Certificates

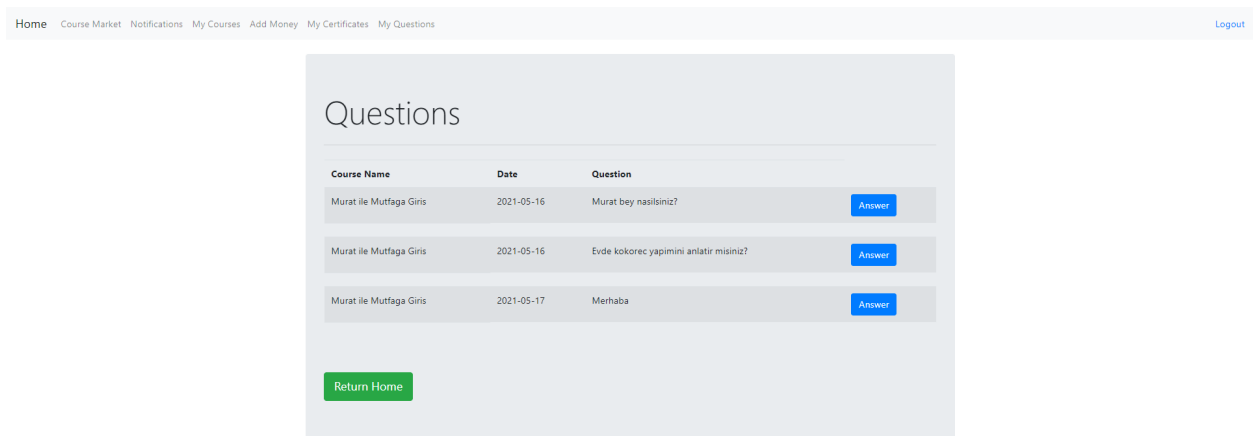


Figure 14: My Questions

Navigation bar at the top of the page directs you to every page at any time. For instance, you can see your certificates by clicking “My Certificates”, and questions by clicking “My Questions”. In “My Questions”, you can see the answer of the instructor by clicking “Answer” button.

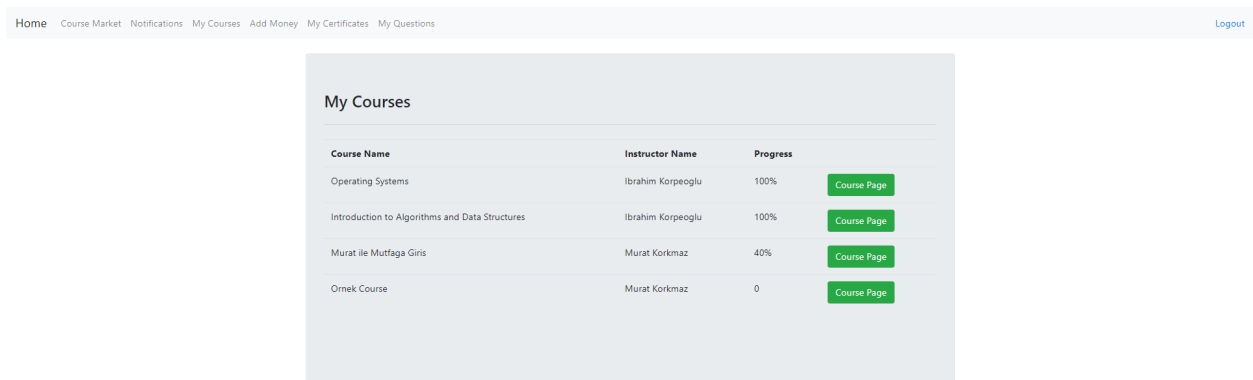


Figure 15: My Courses Page

If you click “My Courses”, you’re directed to the page where you can see the courses you have bought. To go to the course page, you can click the “Course Page” button.

The screenshot shows the 'Course Page' for 'Introduction to Algorithms and Data Structures'. The page has a navigation bar at the top with links: Home, Course Market, Notifications, My Courses, Add Money, My Certificates, My Questions, and Course Page. A 'Logout' link is in the top right corner. On the left, there's an 'Announcements' section with a table of recent updates. The main content area features the course title, a 'Lectures' table, a progress bar, and buttons for 'Give Feedback' and 'Get Certificate'. On the right, there's a sidebar with buttons for 'Request Refund', 'Q&A Page', 'Assignments', and 'Make a Complaint'.

Date	Text
2021-05-16	In the weekend, the assignments have been graded
2021-05-16	Please take the assignments seriously
2021-05-16	You can see the skeleton code, it is in github
2021-05-17	Yeni bir announcement

Lecture Id	Lecture Name	Description	Duration	
0	Linked List	Linked lists	01:00:00	Watch Lecture
1	Array	Arrays	00:34:00	Watch Lecture
2	Graphs		01:14:00	Watch Lecture
3	Trees		00:35:00	Watch Lecture
4	Dynamic Programming	Dynamic programming information	00:36:00	Watch Lecture

Progress

Give Feedback Get Certificate

Request Refund Q&A Page Assignments Make a Complaint

Figure 16: Course Page

The screenshot shows the 'Course Q&A Page' for 'Introduction to Algorithms and Data Structures'. The page has the same navigation bar as Figure 16. The main content area features the course title, a 'Questions' section with a text input field containing 'Hello' and an 'Answer' button, and buttons for 'Return Course' and 'Ask Question'.

Questions

Hello Answer

Return Course Ask Question

Figure 17: Course Q&A Page

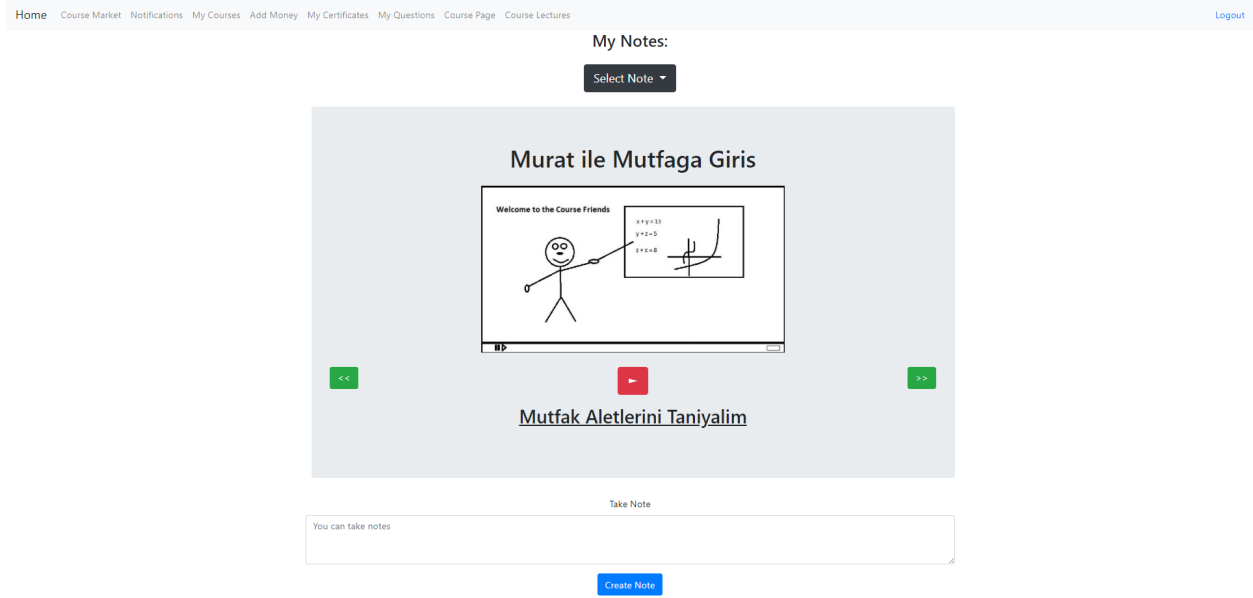


Figure 18: Lecture Watching Page

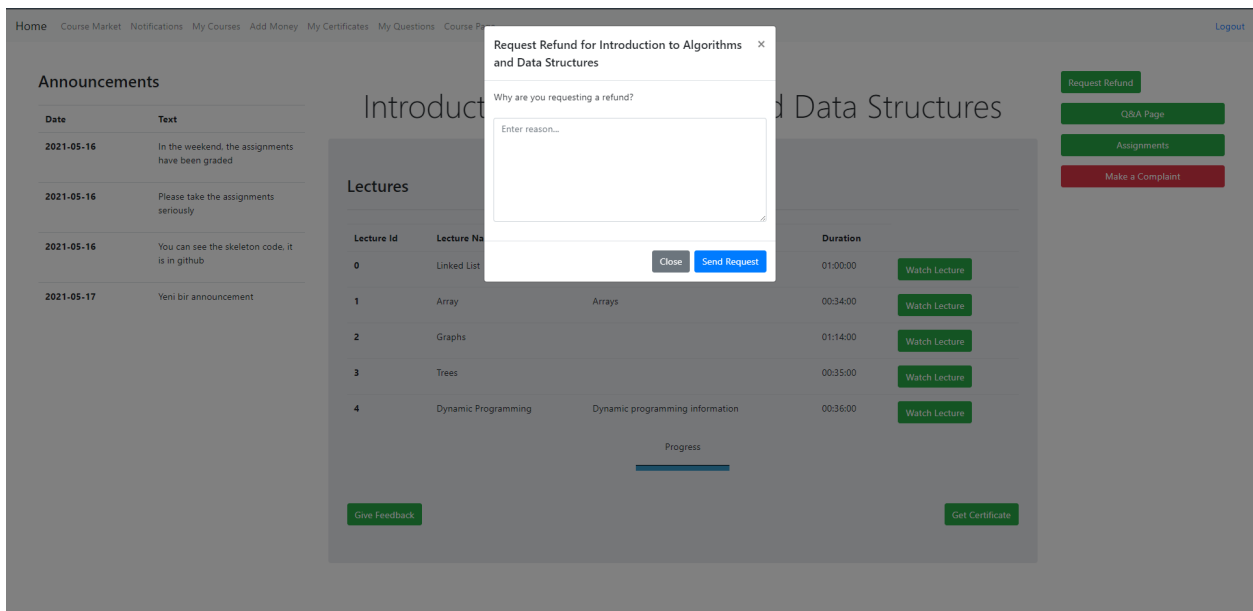


Figure 19: Request Refund as Student

If you go to the course page you have bought, you can see the announcements at the left hand side and lectures in the middle. To watch a lecture, you can click the “Watch Lecture” button, to refund a request, you can click the “Refund Request” button. If you have already requested a refund for a course, this button is inactive. “Q&A Page” button directs you to the questions and answers page where the questions asked and answers given in that course are displayed. In order to be able to get a certificate, you must complete the assignments and get a grade greater than or equal to the threshold from every assignment and you must have a full progress.

Home	Course Market	Notifications	My Courses	Add Money	My Questions	Course Page	Assignments	Logout
------	---------------	---------------	------------	-----------	--------------	-------------	-------------	--------

Assignments Status					
Assignment ID	Assignment	Threshold	Status	Attempt Count	See Attempts
1	Assignment 1	40	Successful	1	See attempts

Figure 20: Assignment Page

[Home](#)
[Course Market](#)
[Notifications](#)
[My Courses](#)
[Add Money](#)
[My Questions](#)
[Course Page](#)
[Assignments](#)
[Attempts](#)
[Logout](#)

Attempts for Assignment 1

Threshold: 40
Please wait for instructor for grading.

Attempt	Grade	Status	Submission Time
1	0	Not graded	2021-05-17 09:25:19

Make Submission

Figure 21: Assignment Attempts Page

You can see the assignments by clicking the “Assignments” button. To see the status of your submission, you can click “See attempts” for the assignment you wish to see. If the assignment is not yet successfully completed, you can make a submission by clicking the “Make Submission” button. If you want to make a complaint to the admin, you can click the “Make a Complaint” button on Course Page.

[Go Back to Your Course](#)

Please explain with details why you are creating a complaint about this course.

Enter Lecture Description

[Make a Complaint](#)

Figure 22: Complaint Page

If you have completed the course (i.e, you have watched every lecture and completed every assignment with a passing grade), you can get your certificate by clicking the Get Certificate button.

[Home](#) [Course Market](#) [Notifications](#) [My Courses](#) [Add Money](#) [My Certificates](#) [My Questions](#) [Course Page](#) [Logout](#)

Your Comments Matter

Your Rating

Your Comment

Return to the Course Page

Send Your Feedback

Figure 23: Send Feedback Page

[Home](#) [Course Market](#) [Notifications](#) [My Courses](#) [Add Money](#) [My Certificates](#) [My Questions](#) [Course Page](#) [Logout](#)

Your Comments Matter

Your Rating: 3.5

Your Comment

It was averagely good

Return to the Course Page

Change Your Feedback

Figure 24: Feedback Page (After Sending Feedback)

If you have watched every lecture of a course, you can give feedback for that course. These feedbacks are seen by other students who visit the page of that course. You can change your feedback anytime.

[Home](#) [Course Market](#) [Notifications](#) [My Courses](#) [Add Money](#) [My Certificates](#) [My Questions](#) [Logout](#)

Refund Request Notifications

Your refund request for Türkiye'de Ne Yenir is Accepted
Your refund request for Murat ile Mutfağa Giris is Rejected

Announcement Notifications

New announcements for Introduction to Algorithms and Data Structures
New announcements for Murat ile Mutfağa Giris

Return Home

Figure 25: Notifications Page

Notifications about your recently issued refund requests and recent announcements about the courses you have enrolled to are shown in the Notifications page which is accessible with Notifications button.

Course Creator User Manual

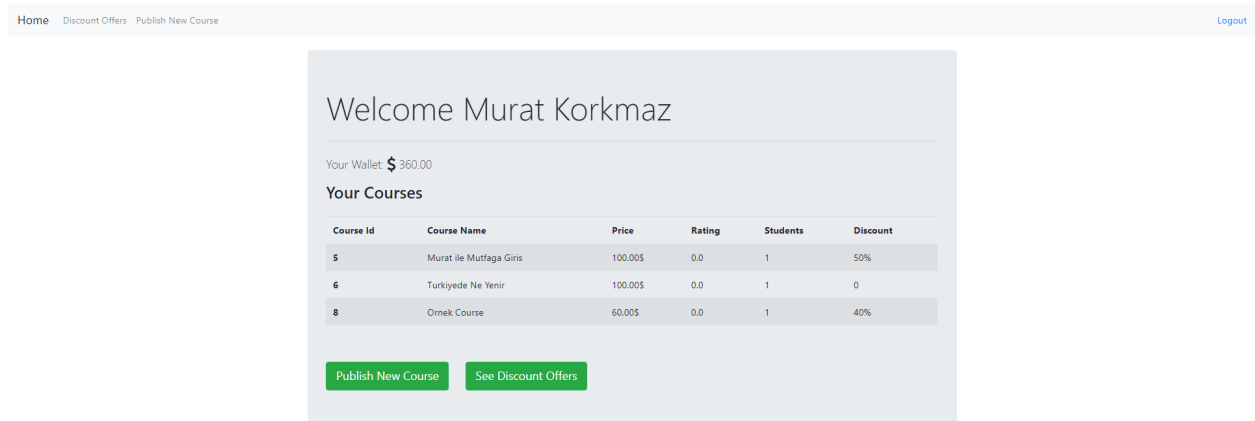


Figure 26: Course Creator Home Page

When you log in to the website, the home page welcomes you with information about your wallet and your courses (if you have created any).

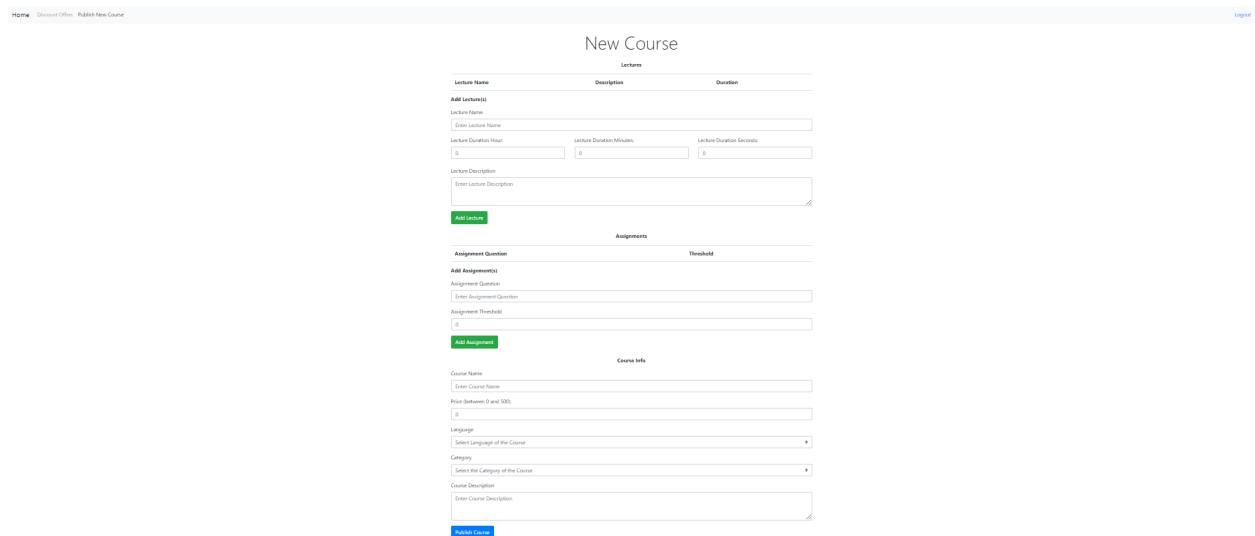


Figure 27: Publish A Course Page

You can publish a course by clicking the “Publish New Course” button or “Publish New Course” link in the navigation bar. In order to publish a course, you must create at least one lecture. Assignment is optional while creating a course. You can go to the course page of your courses by clicking the course in the table in the home page.



Figure 28: Course Page from the Perspective of Course Creator

You can create an announcement easily by writing the text and clicking the “Announce” button. You can see the questions asked by the students by clicking the “Q&A Page” button. You can answer the questions asked on this page.

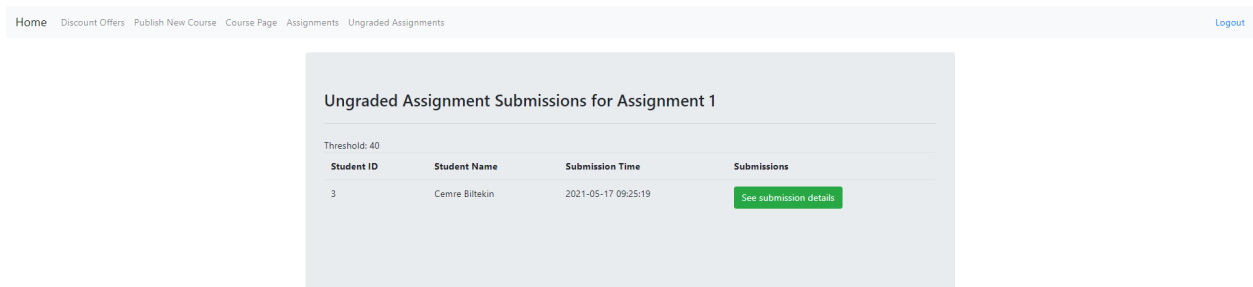


Figure 29: Ungraded Submissions Page

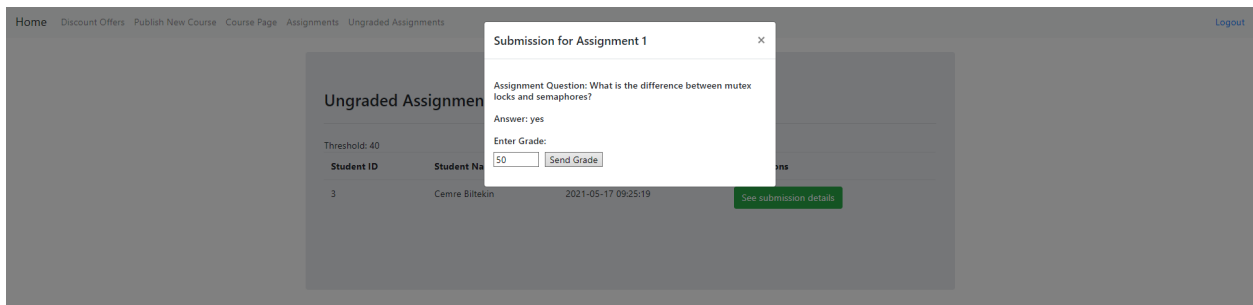


Figure 30: Giving Grade to Student

You can grade the assignments of the students by going to the assignments page by clicking the View Assignments button. You can see ungraded submissions and grade them by clicking on a certain assignment and send the grade you desire.

Discount Offers for Your Courses

Discount Offers							
Course ID	Course Name	Current Price	Percentage	Start Date	End Date	Approve Status	
5	Murat ile Mutfaga Giris	200.00	50	2021-05-17	2021-05-20	Approved	<button>Reject</button>
6	Turkiyede Ne Yenir	100.00	30	2021-05-17	2021-05-22	Not approved	<button>Accept</button>
8	Ornek Course	100.00	40	2021-05-17	2021-05-20	Approved	<button>Reject</button>

Figure 31: Discount Offers Page

If you click the discount offers, you can see the discount offers made by the admin. You must approve the discount offer in order for that discount offer to be applied. That way you can increase the number of students who bought your course. You can also Reject the offer.

Admin User Manual

Welcome Admin Can Alpay

You can check refund requests, offer discounts, see complaints and stats below.

Refund RequestsDiscountsStatsComplaints

Figure 32: Admin Welcome Page

When you login as the administrator, a welcome page (home page) opens. On the page, you can see four buttons. Each button directs you to another page. You can also use the top navigation bar to access these pages that the corresponding buttons open.

Refund Requests			
Refund ID	Student Name	Course Name	Details
5	Mustafa Yasar	Ornek Course	<button>See details</button>

Figure 33: Refund Requests Page

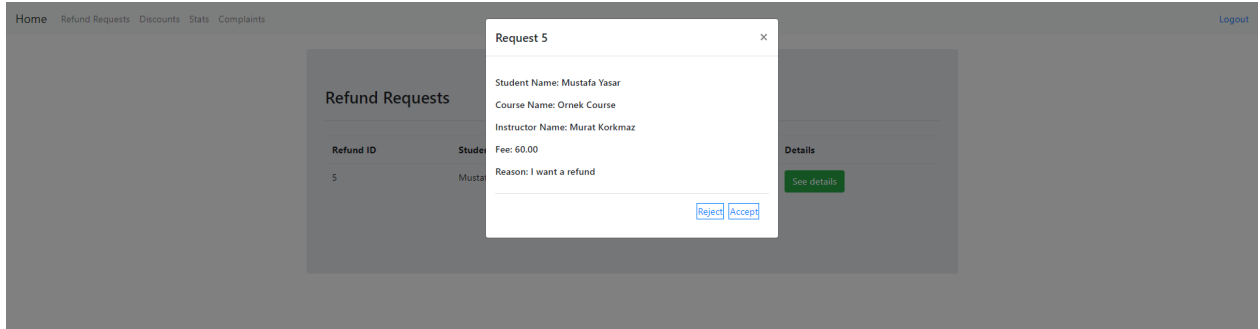


Figure 34: Assessment of Refund Request

When you click on “Refund Requests”, you can access the page where you can see refund requests made by students in a table. To be able to see the details of the request, you must click on the “See details” link on the corresponding row. The modal that opens up has the refund request details like the refund reason. You can reject it via Reject button or accept it via Accept button.

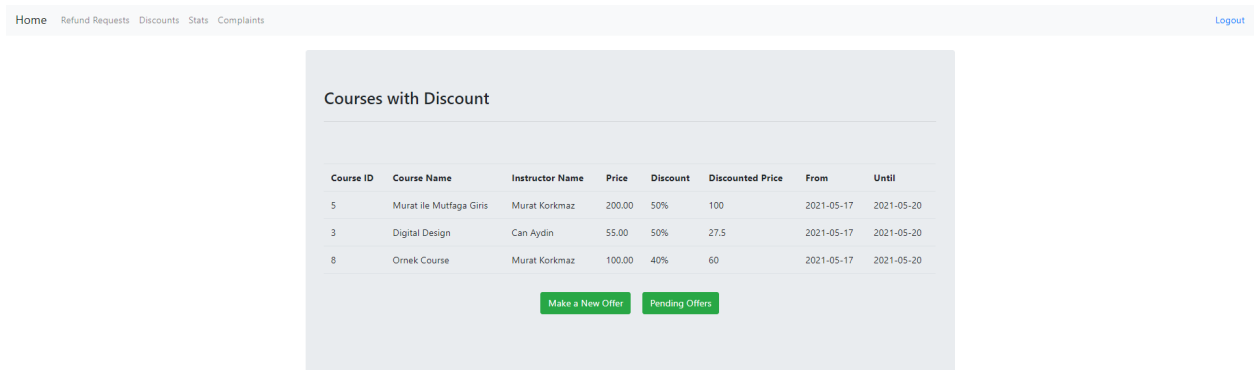


Figure 35: Discounts Page (Ongoing Discounts)

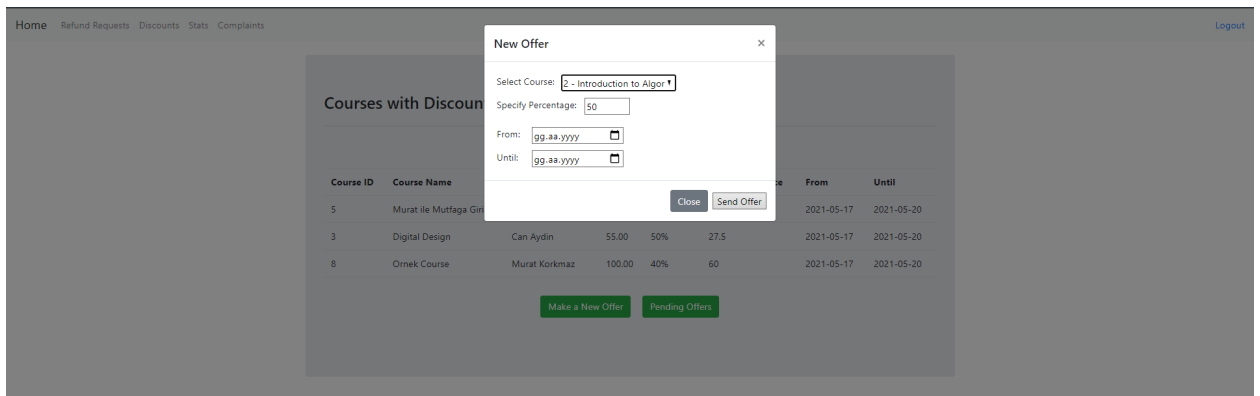


Figure 36: Making a New Offer as Admin

Home	Refund Requests	Discounts	Stats	Complaints	Pending Offers	Logout
----------------------	---------------------------------	---------------------------	-----------------------	----------------------------	--------------------------------	------------------------

Pending Discount Offers								
Discount ID	Course Name	Instructor Name	Price	Discount	Discounted Price	From	Until	Cancel Discount
3	Turkiyede Ne Yenir	Murat Korkmaz	100.00	30%	70	2021-05-17	2021-05-22	Cancel

Figure 37: Pending Discount Offers

When you click on “Discounts”, you can see the current applied discounts on courses and the information about them. You can choose to make a new discount offer for a course by clicking on “Make a New Offer”. You have to specify discount attributes and hit send to send the offer to the corresponding course creator. If you wish to see the pending discount offers that have not yet been assessed by their course creators, you must click on “Pending Offers” to see which discount offers are still waiting to be assessed.

Home	Refund Requests	Discounts	Stats	Complaints	Logout
----------------------	---------------------------------	---------------------------	-----------------------	----------------------------	------------------------

Stats	
	Student Count: 6
	Instructor Count: 3
	Total Courses in the System: 8
	Most Popular Course: Operating Systems
	Best Course: Introduction to Algorithms and Data Structures
	Worst Course: Digital Design
	Best Instructor: Ibrahim Korpeoglu
	Worst Instructor: Can Aydin
	Pending Refund Requests: 1
Go to Home	

Figure 38: Stats Page

When you click on “Stats”, you can see certain statistics data about the application like number of registered students, course creators on Stats page. You can see the most popular course, worst course based on its rating etc. too. This page is an informative report page about the application.

Home	Refund Requests	Discounts	Stats	Complaints	Logout
----------------------	---------------------------------	---------------------------	-----------------------	----------------------------	------------------------

<p>Student: Mustafa Yasar - Course: Murat ile Mutfaga Giris</p> <p>2021-05-17</p> <p>This course is soooo hard</p>
<p>Student: Mustafa Yasar - Course: Turkiyede Ne Yenir</p> <p>2021-05-17</p> <p>Bu kurs sacma</p>

Figure 39: Admin View of Complaints Page

When you click on “Complaints”, you can access the student complaints page where the complaints about the courses by students are shown in detail.

Website

<https://oguzkaanimamoglu.github.io/Online-Course-Platform/>