



# Bilkent Üniversitesi

## Database Systems

Department of Computer Engineering

Spring 2021

## Final Report

Online Course Platform – Sapientia

Project Group 4:

- Ömer Ünlüsoy - Section 01 - 21702136
- Elif Gamze Güliter - Section 01 - 21802870
- İrem Tekin - Section 01 - 21803267
- Ece Ünal - Section 01 - 21703149

Supervisor:

- Uğur Gündükay

## Table of Contents

<b>1. Introduction.....</b>	<b>4</b>
<b>1.1 Contribution of Each Group Member.....</b>	<b>5</b>
<b>2. Final E/R Diagram.....</b>	<b>6</b>
<b>3. Final Relation Schemas .....</b>	<b>7</b>
<b>3.1. Student .....</b>	<b>7</b>
<b>3.2. Instructor .....</b>	<b>7</b>
<b>3.3. Admin .....</b>	<b>7</b>
<b>3.4. Course .....</b>	<b>7</b>
<b>3.5. Section .....</b>	<b>7</b>
<b>3.6. Lecture .....</b>	<b>8</b>
<b>3.7. Quiz.....</b>	<b>8</b>
<b>3.8. Quiz_Question .....</b>	<b>8</b>
<b>3.9. Note .....</b>	<b>9</b>
<b>3.10. Course_Announcement.....</b>	<b>9</b>
<b>3.11. QnA_Entry_Student .....</b>	<b>9</b>
<b>3.12. QnA_Entry_Instructor .....</b>	<b>10</b>
<b>3.13. Complaint_Entry_Student .....</b>	<b>10</b>
<b>3.14. Complaint_Entry_Instructor .....</b>	<b>10</b>
<b>3.15. Wishlist .....</b>	<b>11</b>
<b>3.16. Enrolls .....</b>	<b>11</b>
<b>3.17. Certificate.....</b>	<b>11</b>
<b>3.18. Teaches .....</b>	<b>11</b>
<b>3.19. Evaluates_Complaint_Entry_Student.....</b>	<b>12</b>

<b>3.20.</b>	<b>Evaluates_Complaint_Entry_Instructor.....</b>	<b>12</b>
<b>3.21.</b>	<b>Discount .....</b>	<b>12</b>
<b>3.22.</b>	<b>Zoom_Session.....</b>	<b>13</b>
<b>3.23.</b>	<b>Take_Quiz.....</b>	<b>13</b>
<b>3.24.</b>	<b>Take_Quiz_Question .....</b>	<b>13</b>
<b>3.25.</b>	<b>Take_Lecture .....</b>	<b>13</b>
<b>4.</b>	<b><i>Implementation Details</i> .....</b>	<b>15</b>
<b>4.1.</b>	<b>Languages .....</b>	<b>15</b>
<b>4.2.</b>	<b>Database and Connection .....</b>	<b>15</b>
<b>4.3.</b>	<b>Implementation .....</b>	<b>15</b>
<b>5.</b>	<b><i>Advance Database Features</i>.....</b>	<b>16</b>
<b>5.1.</b>	<b>Student Main Page Course Query.....</b>	<b>16</b>
<b>5.2.</b>	<b>Login Page Course Query .....</b>	<b>16</b>
<b>5.3.</b>	<b>Admin Refund Request Query.....</b>	<b>17</b>
<b>5.4.</b>	<b>Certificate Query .....</b>	<b>17</b>
<b>5.5.</b>	<b>Lecture Content Order Query.....</b>	<b>18</b>
<b>6.</b>	<b><i>User Manual</i>.....</b>	<b>19</b>
<b>6.1.</b>	<b>Instructor User Manual.....</b>	<b>19</b>
<b>6.2.</b>	<b>Student User Manual.....</b>	<b>34</b>
<b>6.3.</b>	<b>Admin User Manual.....</b>	<b>53</b>
<b>7.</b>	<b><i>Website</i> .....</b>	<b>65</b>
<b>8.</b>	<b><i>Appendix</i> .....</b>	<b>66</b>
<b>8.1.</b>	<b>SQL Create Table Code.....</b>	<b>66</b>

## 1. Introduction

In this project, we implemented a web-based course platform similar to Udemy. The purpose of the platform is to enable instructors to build their online courses as well as students to reach out to these contents and educate themselves. The name of the platform is **Sapientia** which means **wisdom** in Latin. In Sapientia, instructors can create courses about the topics that they are proficient about and students can purchase these courses in order to enroll. A course consists of several sections that have lectures as well as quizzes. Students can watch the lectures and solve quizzes. Students receive a certificate indicating that they have finished the online course if only if they watch all of the lectures and get at least the threshold value that is decided by the instructor as their quiz averages. In addition, Sapientia consists of students' wishlist, students' notes for a specific lecture, Q&A section for students and instructors to interact with each other, ratings of courses, discount or refund for courses, and at last, announcements made by instructors about a course.

## **1.1 Contribution of Each Group Member**

### **1.1.1. Ömer Ünlüsoy**

Contributed to E/R Diagram design. Designed the relational schemas. Designed the SQL create table queries. Implemented several data manipulation queries including search and filter. Implemented several pages like contents, view lecture, view quiz.

### **1.1.2. İrem Tekin**

Contributed to E/R diagram, SQL queries for UI's and mockup designs in design report, UI design, data manipulation with SQL queries, general structure (navigation etc.) of the platform. Implemented most of the pages for student, instructor and admin with PHP and HTML and edited some CSS files for the pages.

### **1.1.3. Gamze Güliter**

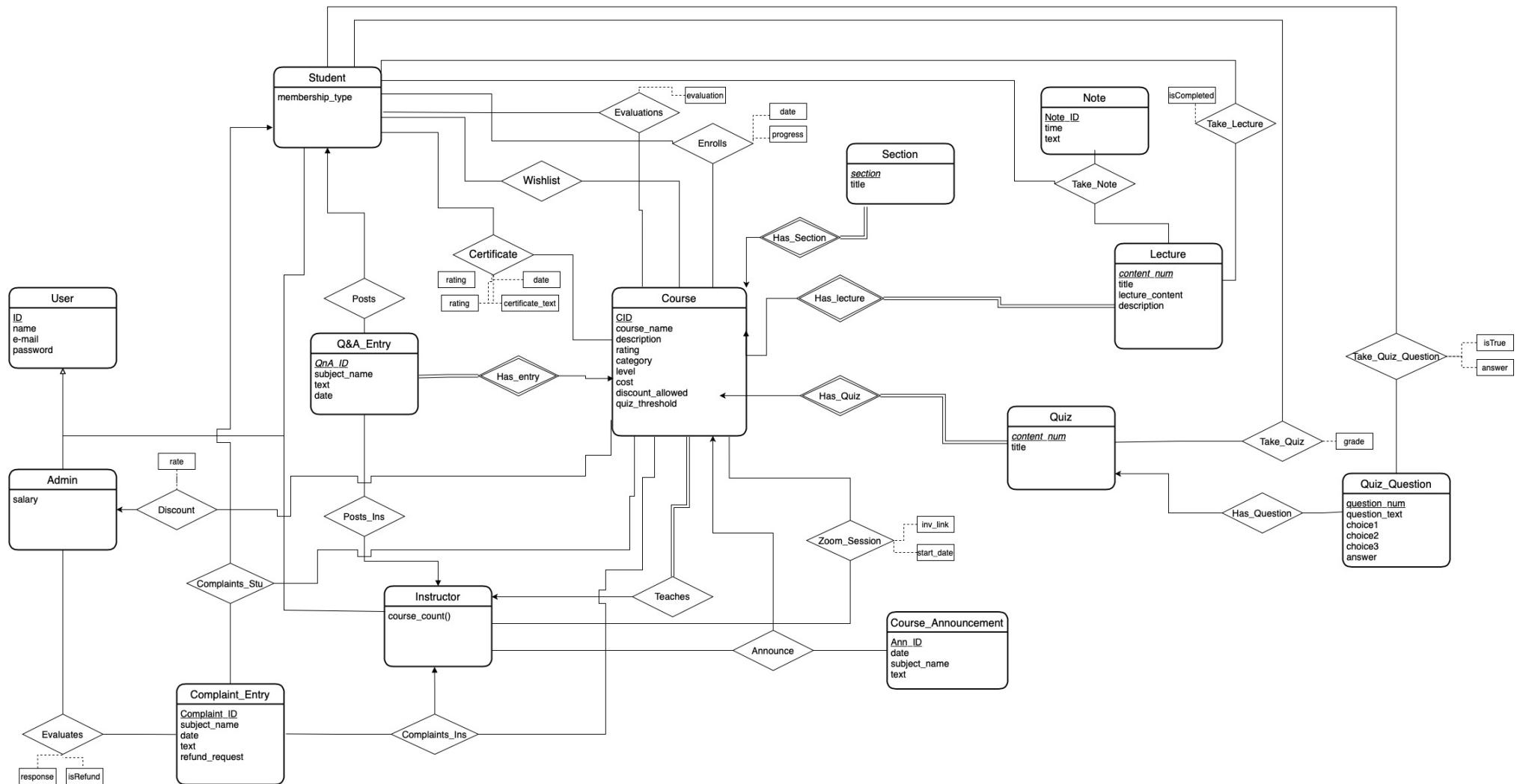
I have worked on Graphical User Interface most of the time, I implemented most of the pages with the help of İrem and Ece. I also helped my friends on back-end side. I assisted them on implementation of php and debugging process.

### **1.1.4. Ece Ünal**

Contributed to E/R Diagram design. Helped to implement SQL queries, and UI design. Contributed to implementation of most of the front-end part. Contributed to debugging process for the PHP and HTML codes. Edited some CSS files for front-end to achieve a more user friendly design.

## 2. Final E/R Diagram

E-R Diagram of Sapientia



Note: Used software does not have a dashed underline so weak entities are marked with italic underline for this E/R.

### **3. Final Relation Schemas**

#### **3.1. Student**

Student(SID, name, e\_mail, password, membership\_type)

#### **3.2. Instructor**

Instructor(IID, name, e\_mail, password)

#### **3.3. Admin**

Admin(AID, name, e\_mail, password, salary)

#### **3.4. Course**

Course(CID, IID, course\_name, description, rating, category, level, cost, discount\_allowed, quiz\_threshold)

Foreign Key: IID REFERENCES Instructor(IID)

Unique: (IID, course\_name)

#### **3.5. Section**

Section(CID, section, title)

Foreign Key: CID REFERENCES Course(CID)

Index: section

### **3.6. Lecture**

Lecture(CID, section, content\_num, title, lecture\_content, description)

Foreign Key: (CID, section) REFERENCES Section(CID, section)

Unique: (CID, title)

Index: section

Index: content\_num

### **3.7. Quiz**

Quiz(CID, section, content\_num, IID, section, title)

Foreign Key: (CID, section) REFERENCES Section(CID, section)

Foreign Key: IID REFERENCES Instructor(IID)

Unique: (CID, title)

Index: section

Index: content\_num

### **3.8. Quiz\_Question**

Quiz\_Question(CID, section, content\_num, question\_num, question\_text, choice1, choice2, choice3, answer)

Foreign Key: (CID, section, content\_num) REFERENCES Quiz(CID, section, content\_num)

### **3.9. Note**

Note(Note\_ID, SID, section, content\_num, time, text)

Foreign Key: SID REFERENCES Student(SID)

### **3.10. Course\_Announcement**

Course\_Announcement(Ann\_ID, IID, CID, date, subject\_name, text)

Foreign Key: IID REFERENCES Instructor(IID)

Foreign Key: CID REFERENCES Course(CID)

### **3.11. QnA\_Entry\_Student**

QnA\_Entry\_Student(QnA\_ID\_S, SID, CID, subject\_name, text, date)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: CID REFERENCES Course(CID)

### **3.12. QnA\_Entry\_Instructor**

QnA\_Entry\_Instructor(QnA\_ID, QnA\_ID\_S, IID, CID, subject\_name, text, date)

Foreign Key: QnA\_ID\_SREFERENCES QnA\_Entry\_Student(QnA\_ID\_S)

Foreign Key: IID REFERENCES Instructor(IID)

Foreign Key: CID REFERENCES Course(CID)

### **3.13. Complaint\_Entry\_Student**

Complaint\_Entry\_Student(Complaint\_ID, SID, CID, subject\_title, text, date, refund\_request)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: CID REFERENCES Course(CID)

### **3.14. Complaint\_Entry\_Instructor**

Complaint\_Entry\_Instructor(Complaint\_ID, SID, CID, subject\_name, text, date)

Foreign Key: IID REFERENCES Instructor(IID)

Foreign Key: CID REFERENCES Course(CID)

### **3.15. Wishlist**

Wishlist(SID, CID)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: CID REFERENCES Course(CID)

### **3.16. Enrolls**

Enrolls(SID, CID, progress, date)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: CID REFERENCES Course(CID)

### **3.17. Certificate**

Certificate(SID, CID, certificate\_text, rating, comment, date)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: CID REFERENCES Course(CID)

### **3.18. Teaches**

Teaches(IID, CID)

Foreign Key: IID REFERENCES Instructor(IID)

Foreign Key: CID REFERENCES Course(CID)

### **3.19. Evaluates\_Complaint\_Entry\_Student**

Evaluates\_Complaint\_Entry\_Student(AID, Complaint\_ID, response, isRefund)

Foreign Key: AID REFERENCES Admin(AID)

Foreign Key: IID Complaint\_ID REFERENCES Complaint\_Entry\_Student(Complaint\_ID)

### **3.20. Evaluates\_Complaint\_Entry\_Instructor**

Evaluates\_Complaint\_Entry\_Instructor(AID, Complaint\_ID, response)

Foreign Key: AID REFERENCES Admin(AID)

Foreign Key: Complaint\_ID REFERENCES Complaint\_Entry\_Instructor(Complaint\_ID)

### **3.21. Discount**

Discount(AID, CID, rate)

Foreign Key: AID REFERENCES Admin(AID)

Foreign Key: CID REFERENCES Course(CID)

### **3.22. Zoom\_Session**

Zoom\_Session(Session\_ID, IID, CID, invite\_link, start\_date)

Foreign Key: IID REFERENCES Instructor(IID)

Foreign Key: CID REFERENCES Course(CID)

### **3.23. Take\_Quiz**

Take\_Quiz(SID, CID, section, content\_num, grade)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: (CID, section, content\_num) REFERENCES Quiz(CID, section, content\_num)

### **3.24. Take\_Quiz\_Question**

Take\_Quiz\_Question(SID, CID, section, content\_num, question\_num, answer, isTrue)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: (CID, section, content\_num, question\_num) REFERENCES Quiz(CID, section, content\_num, question\_num)

### **3.25. Take\_Lecture**

Take\_Quiz\_Lecture(SID, CID, section, content\_num, isCompleted)

Foreign Key: SID REFERENCES Student(SID)

Foreign Key: (CID, section, content\_num) REFERENCES Lecture(CID, section, content\_num)

## **4. Implementation Details**

### **4.1. Languages**

Four different languages have been used in this project. The first language is SQL which is used for database definition and modification. The second language is PHP. It is used to send queries to the database at the server side. The connection to the database has been established by PHP. The third language is HTML which is used to create Web user interface. HTML is selected due to its compatibility with PHP. Both PHP and HTML code is located at .php files and they can send information to each other which eases the coding. Last language is CSS, a style sheet language helped project to look pretty. Most HTML and CSS code have been produced by an application called Nicepage. IntelliJ and PHPstorm IDEs are used.

### **4.2. Database and Connection**

MySQL has been used in this project for both database definition and modification. The database is located at one member's Dijkstra account so that every member can access. The connection is established by a PHP script.

### **4.3. Implementation**

First, the E/R diagram and schemas are designed as the first stage of the implementation. SQL create table queries are designed and tables are created at a Dijkstra account. Main page structure of the website is determined and SQL database manipulation queries are designed for each page. Then, page user interfaces are implemented with HTML and CSS. At this point, the app created static HTML and CSS codes which was a major challenge for the UI part.

After implementing the UI, HTML files are turned into PHP files and UI components like buttons and text fields, are connected. After then, SQL manipulation queries are implemented in PHP files and each query is tested with several cases. Finally, page and session connections are implemented.

Gamze Güliter and Ece Ünal were mainly responsible from the web UI (HTML and CSS) while Ömer Ünlüsoy and İrem Tekin were mainly responsible for SQL queries and PHP. However, every group member has worked at every stage of the implementation.

## 5. Advance Database Features

### 5.1. Student Main Page Course Query

```
select *
from Discount right join Course on Course.CID = Discount.CID and
rate between $discount_min and $discount_max
natural join Instructor
where
cost between $price_min and $price_max
and category = $category and level = $level
and (course_name like %$search_word% or Instructor.name like %$search_word%)
limit $page_num, $page_entry_num;
```

IID	AID	CID	rate	CID	course_name	description		
1	1	1	20.00	1	Music Theory for Beginners	This course aims to teach the basics of music theory to newco...		
rating	category	level	cost	discount_allowed	quiz_threshold	name	e_email	password
95.00	Music	Beginner	30.99	0	8.00	Irem Tekin	irem@tekin	1234

Figure 1: Example run with rate between 0, 20; cost between 10, 50; category = Music; level = Beginner; search word = music

This query shows all the courses that satisfy student's criteria at the student main page.

Everything starting with \$ sign is a variable taken from the student. This query joins three tables; uses price, level, and discount filters. It also implements search feature with “like” phrase. Finally, it uses “limit” to limit the number of Courses listed at each page.

### 5.2. Login Page Course Query

```
select *
from Discount right join Course on Course.CID = Discount.CID
natural join Instructor
order by rate DESC
limit 0, 5;
```

	IID	AID	CID	rate	CID	course_name	description
1	1	1	2	20.00	2	Introduction to Badminton	This course aims to introduce badminton basics.
2	1	1	1	20.00	1	Music Theory for Beginners	This course aims to teach the basics of music theory to newcomers.
3	1	<null>	<null>	<null>	4	SQL 2	SQL 2
4	3	<null>	<null>	<null>	12	CS 353	database course:)
5	1	<null>	<null>	<null>	7	Linear Algebra for Beginners	Linear Algebra for Beginners course

	rating	category	level	cost	discount_allowed	quiz_threshold	name	e_mail	password
newcomers.	0.00	Sports	Beginner	12.99	1	6.00	Irem Tekin	irem@tekin	1234
	95.00	Music	Beginner	30.99	0	8.00	Irem Tekin	irem@tekin	1234
	0.00	Computer Science	Intermediate	25.00	0	7.00	Irem Tekin	irem@tekin	1234
	0.00	Gardening	Intermediate	80.00	0	7.00	Mustafa Cavdar2	cavdar@cavdar2	1234
	0.00	Math	Beginner	45.89	1	10.00	Irem Tekin	irem@tekin	1234

Figure 2: Example run

This query lists top five courses according to their discount rate at the login page.

### 5.3. Admin Refund Request Query

```
select count(*) as refund_count
from Complaint_Entry_Student
```

	refund_count
	1

Figure 3: Example run with CID = 3

```
where CID = '$CID' and refund_request = 1;
```

This query counts the number of refund request made by a specific student for admin.

### 5.4. Certificate Query

```
select *
from (
  select course_name, CID
    from Course
) as C_subquery
```

```
natural join Certificate
where SID = '$SID';
```

	CID	course_name	SID	certificate_text	rating	comment	date
1	1	Music Theory for Beginners	1	1 <null>	95.00	well designed :)	2021-05-17
2	13	Introduction to Living with a Cat	1 <null>	10.00	<null>	<null>	<null>

Figure 4: Example run with SID = 1

This nested query shows every course, particular student has finished and got certificate for.

## 5.5. Lecture Content Order Query

```
select *
from Lecture
where CID = '$CID' and section = '$section_name'
```

	CID	section	content_num	title	lecture_content	description
1	1	1	1	Welcome to the Music Theory	upload/trial.mp4	WELCOME NEWCOMERS
2	1	1	2	Rhythm Theory	upload/trial.mp4	This lecture explains the Rhythm Theory

Figure 5: Example run with CID = 1 and section = 1

```
order by content_num;
```

This query lists every lecture in a particular section of a course and orders according to their content number in the section.

## 6. User Manual

### 6.1. Instructor User Manual

#### 6.1.1. Instructor Log In

Instructors will login to the Sapientia website by the Login Page.

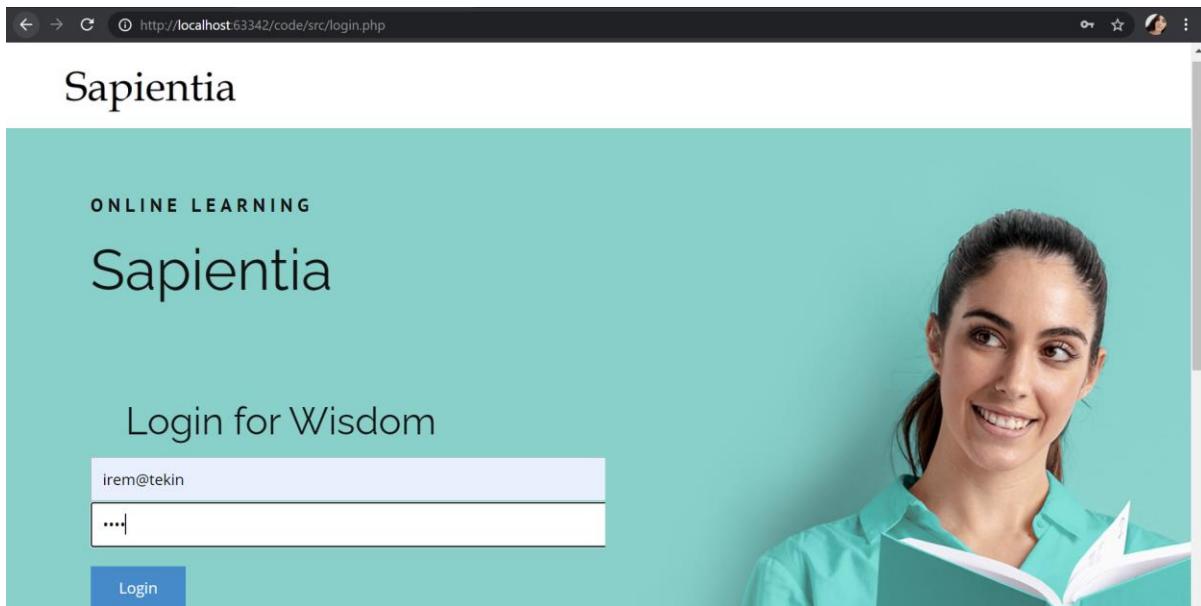


Fig. 6. Instructor Login

#### 6.1.2. Instructor's Main Page

After log in, instructors will be directed to the main page which lists all the courses that the instructor teaches. In order to see this page, the instructor can also use the “**My Courses**” button in the navigation bar.

The screenshot shows a web browser window with the URL [http://localhost:63342/code/src/instructor\\_main\\_courses.php#](http://localhost:63342/code/src/instructor_main_courses.php#). The page title is "Sapientia". The top navigation bar includes links for "Account", "Forum", "My Courses", "My Announcements", and "Logout". A large red "Add Course" button is centered above a table listing two courses. The table has columns for "Course Name", "Category", and "Level". The first course is "Music Theory for Beginners" in the "Music" category at the "Beginner" level, with buttons for "EDIT", "STATISTICS", "COMPLAINT", and "ANNOUNCE". The second course is "Introduction to Badminton" in the "Sports" category at the "Beginner" level, also with "EDIT", "STATISTICS", "COMPLAINT", and "ANNOUNCE" buttons. A vertical scroll bar is visible on the right side of the page.

Course Name	Category	Level
Music Theory for Beginners	Music	Beginner
Introduction to Badminton	Sports	Beginner

Fig. 7. Instructor's Main Page

#### 6.1.3. Adding a Course to the Sapientia Website

In order to add a course, instructor should press the “**Add a Course**” button on the main page. This button will direct the instructor to the “**Add a Course**” page. In Add a Course page, the instructor should specify the name of the course that he/she will teach, category, the level of course, brief description about the course, its price and quiz threshold. Quiz threshold indicates the min grade that a student needs to collect from the quizzes in order to get a certificate from the course.

Add a Course

Course Name	Introduction to Living with a Cat
Category	Psychology
Level	Beginner
This course is for cat lovers.	
Price (\$)	40
Min. Quiz Average (.../10)	7

Fig.8. Adding a course to the Sapientia website

	Science						
SQL 3	Computer Science	Advanced		<a href="#">EDIT</a>	<a href="#">STATISTICS</a>	<a href="#">COMPLAINT</a>	<a href="#">ANNOUNCE</a>
Immanuel Kant	Philosophy	Intermediate		<a href="#">EDIT</a>	<a href="#">STATISTICS</a>	<a href="#">COMPLAINT</a>	<a href="#">ANNOUNCE</a>
Linear Algebra for Beginners	Math	Beginner		<a href="#">EDIT</a>	<a href="#">STATISTICS</a>	<a href="#">COMPLAINT</a>	<a href="#">ANNOUNCE</a>
HTML Basics	Graphic Design	Beginner		<a href="#">EDIT</a>	<a href="#">STATISTICS</a>	<a href="#">COMPLAINT</a>	<a href="#">ANNOUNCE</a>
Introduction to Living with a Cat	Psychology	Beginner		<a href="#">EDIT</a>	<a href="#">STATISTICS</a>	<a href="#">COMPLAINT</a>	<a href="#">ANNOUNCE</a>

Fig.9. Instructor's main page after adding a course with the name "Introduction to Living with Cats"

#### 6.1.4. Editing an Added Course

In order to edit a course, the instructor should press the “Edit” button next to the course name in the main page. Editing a course includes uploading lecture videos, uploading quizzes, changing its name, price, quiz threshold, level, description etc.

As an **additional feature**, the instructor can schedule a zoom meeting with the enrolled students by giving a date and a zoom link. In order to schedule a zoom meeting, the instructor needs to use “Edit a Course” page.

The screenshot shows a web browser window titled "Edit a Course". On the left, there are several input fields for course details:

- Course Name: Introduction to Living with a Cat
- Category: Psychology
- Level: Beginner
- Description: This course is for cat lovers. We all love cats!!
- Price (\$): 30
- Min. Quiz Average (.../10): 7.00
- Zoom Link: (empty)

On the right, there are six pink rounded rectangular buttons with white icons and text:

- ADD SECTIONS (+ icon)
- ADD LECTURE (+ icon)
- ADD QUIZ (+ icon)
- INVITE INSTRUCTORS (person icon)
- EDIT LECTURES (pencil icon)
- EDIT QUIZIEZ (pencil icon)

The screenshot shows a web-based course editing interface. On the left, there are several input fields for course metadata:

- Description:** A text area containing the text "This course is for cat lovers. We all love cats!!".
- Price (\$):** An input field with the value "30".
- Min. Quiz Average (.../10):** An input field with the value "7.00".
- Zoom Link:** A text area containing a Zoom link: "https://us02web.zoom.us/postattendee?id=43".
- Date:** A date input field set to "18.05.2021".
- Time:** A time input field set to "12.00".

On the right side, there are four pink rounded rectangular buttons with white icons and text:

- ADD QUIZ** (with a plus sign icon)
- INVITE INSTRUCTORS** (with a user icon)
- EDIT LECTURES** (with a pen icon)
- EDIT QUIZIEZ** (with a pen icon)

At the bottom center is a large red button labeled "Edit Course" with a hand cursor icon.

Fig 10 and 11. Edit a Course Page

#### 6.1.5 Adding Sections to a Course

In Sapientia, instructors should add sections to their courses, then, add lectures and quizzes. In order to add a section to the course, the instructor should use the “**Add a Section**” button in the course’s edit page. While adding a section, instructor should specify a name and a number for the section.

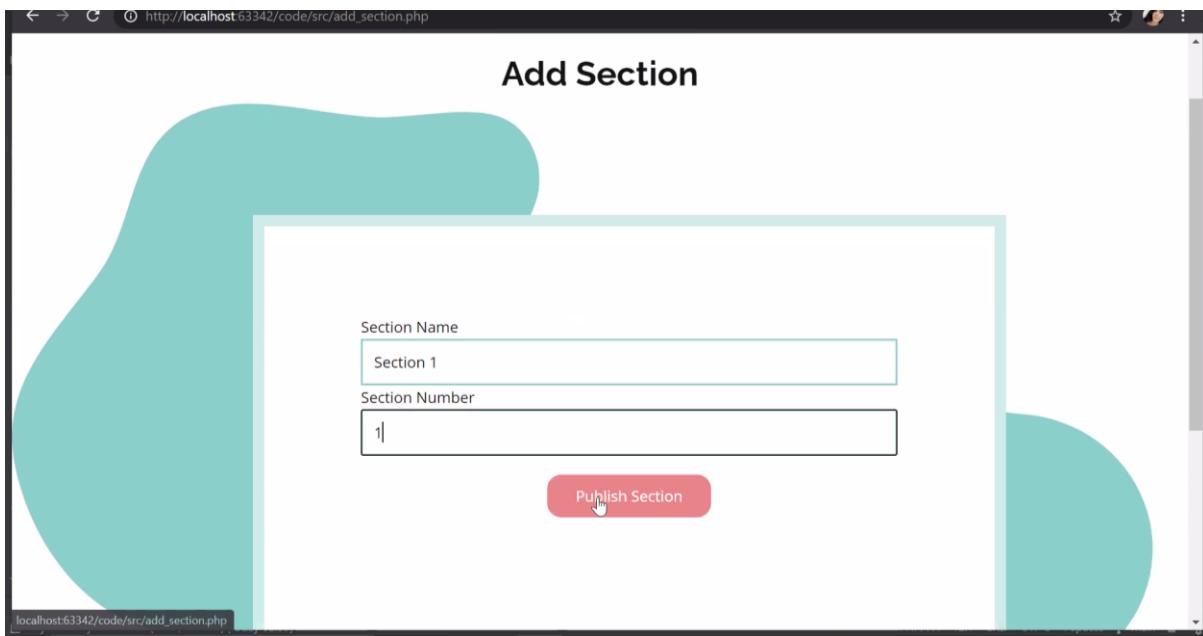


Fig.12 Adding a Section to the Course

#### 6.1.6 Adding Lectures to the Course

In order to add a lecture to the course, an instructor first needs to add at least one section to the course. Following that, in order to add a lecture, the instructor should press the "**Add a Lecture**" button in the edit page. The button will direct the instructor to the "Add a Lecture" page. From that page, the teacher must specify a name, a number, brief description, and a section for the lecture.

In order to upload a lecture video, the instructor must press to the "**Submit Video**" button on the right corner and upload a lecture video from the local computer (See Figures 15,16)

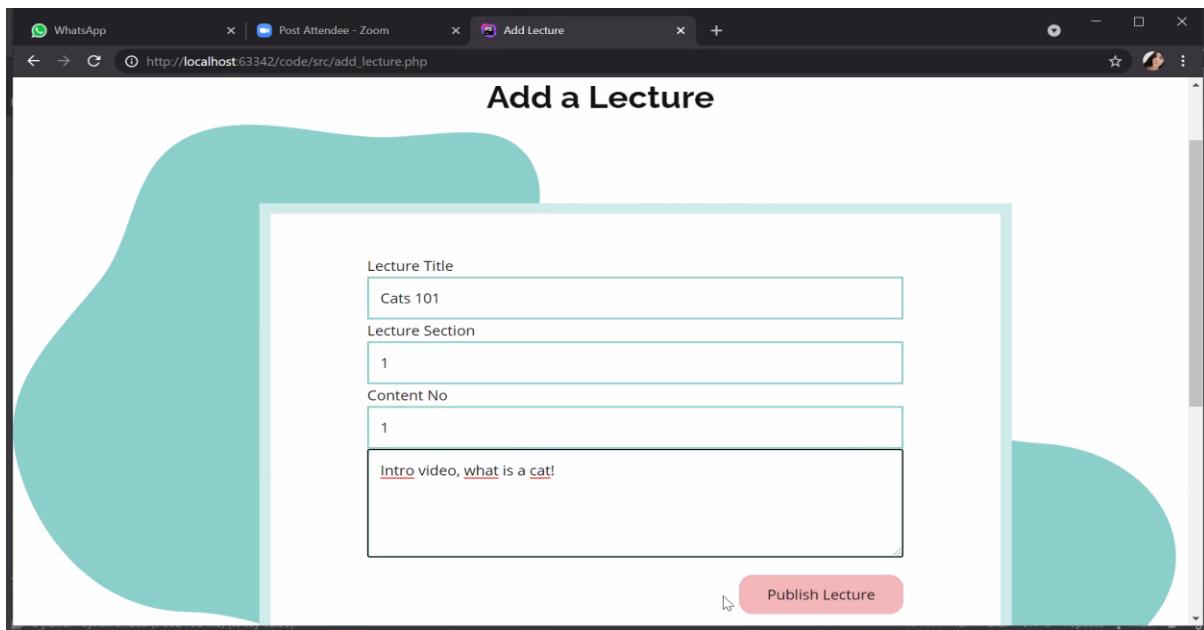


Fig. 13. Add a Lecture Page



Fig.14 Submit Video Button

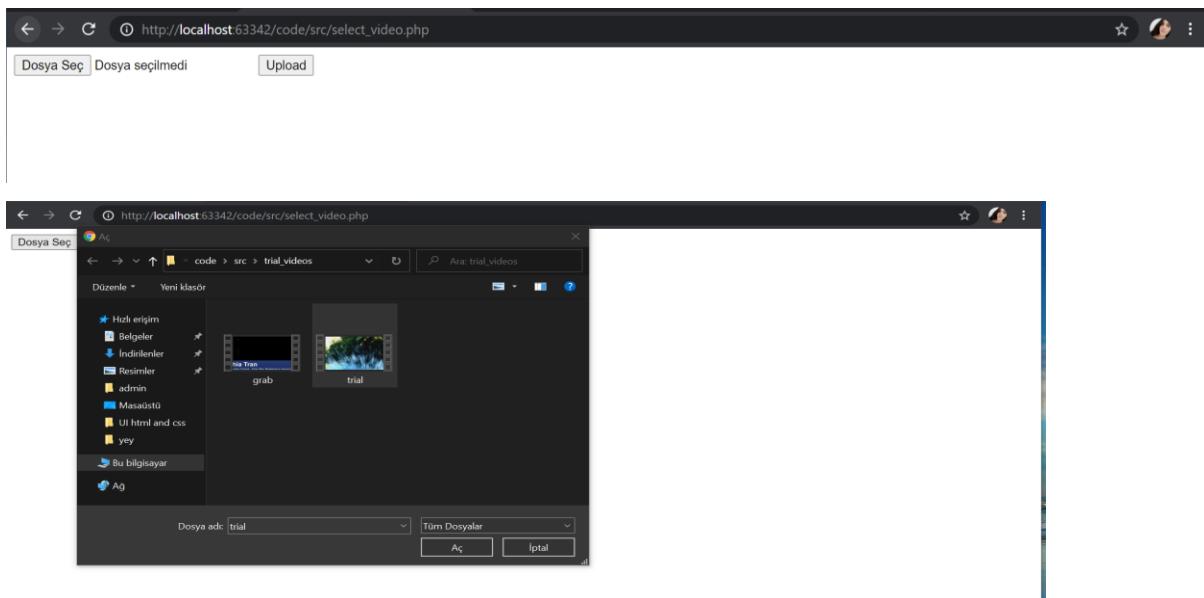


Fig 15 and 16. Uploading a Lecture Video from Local Computer

### 6.1.7 Adding Quizzes to the Course

In order to add a course to the course, an instructor first needs to add at least one section to the course. Following that, in order to add a quiz, the instructor should press the “**Add a Quiz**” button in the edit page. The button will direct the instructor to the “Add a Quiz” page. From that page, the teacher must specify a name, a number and a section for the quiz.

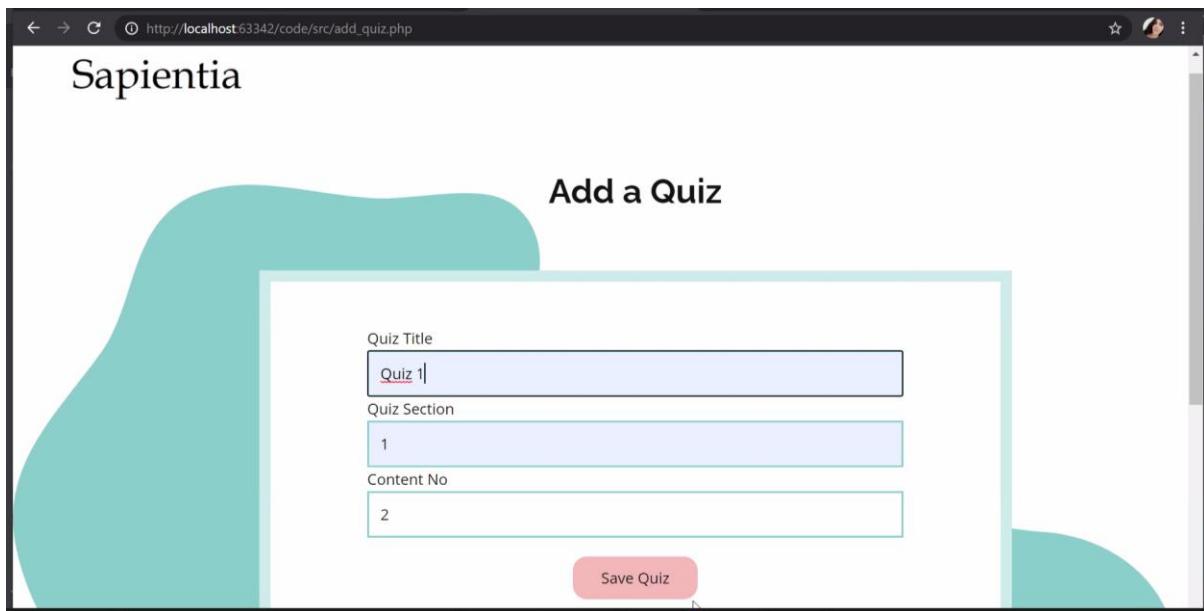


Fig .17 Adding a Quiz

After adding a quiz, instructor should add questions to that quiz. In “**Add a Quiz Question**” page, instructor should write a question, three options and indicate the correct answer of the question. Following that, the instructor can either continue adding questions or finish adding by pressing to the “**Finish**” button.

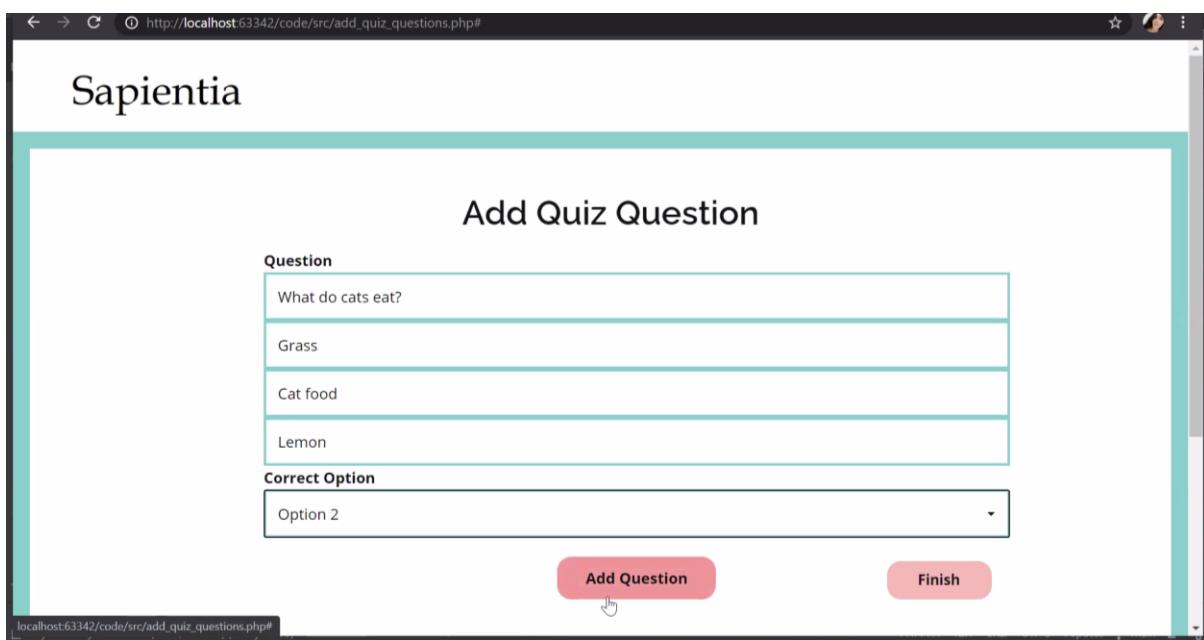


Fig 18. Adding Questions to a Quiz

### 6.1.8 Inviting an Instructor to a Course

As an **additional feature**, instructors can invite other instructors to the course as content creators. In order to do this, the instructor should press the “**Invite an Instructor**” button on the “**Edit Course**” page. The button will direct the instructor to the page shown below. From there, the instructor should give the name and email address of the other instructor in order to add him/her as the content creator.

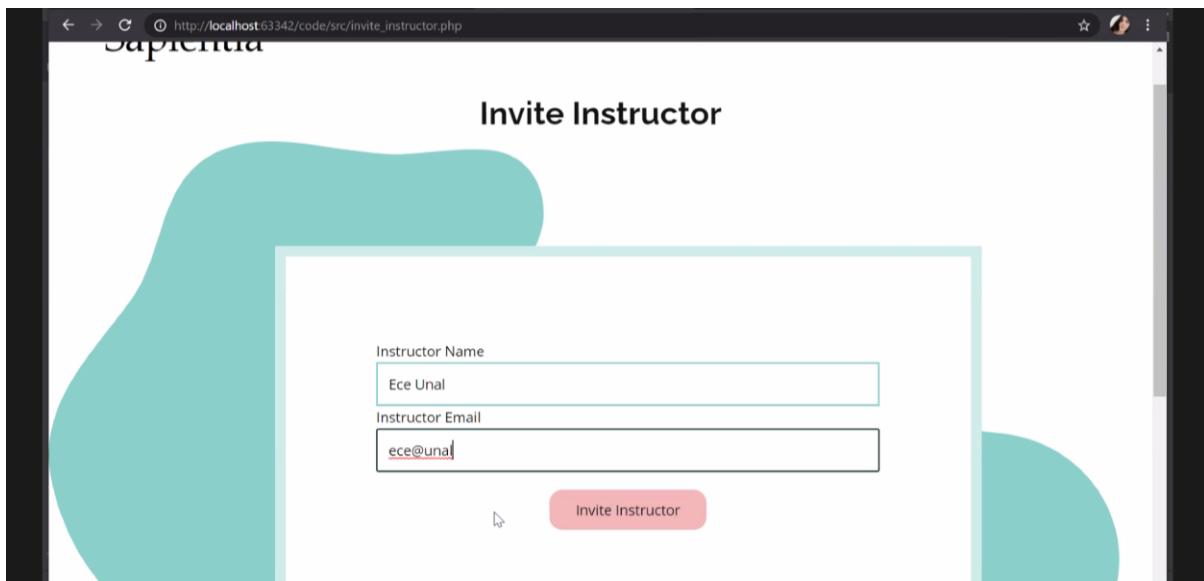


Fig 19. Inviting Another Instructor to the Course.

### 6.1.9 Filling a Complaint About a Course

An instructor can add a complaint such as a technical issue about his/her course to the admin. In order to fill a complaint, he/she needs to press to the “**Fill a Complaint**” button which is next to the course’s name in the main page. From this page, the instructor must specify a subject, date and a brief description about the complaint and press to the “**Send Complaint**” button.

The screenshot shows a web page with a light gray background. At the top, there is a navigation bar with icons for back, forward, and search, followed by the URL [http://localhost:63342/code/src/instructor\\_fill\\_complaint.php#](http://localhost:63342/code/src/instructor_fill_complaint.php#). Below the URL is a large, bold title "Fill a Complaint". The main content area has a white background with rounded corners and a thin red border. It contains three input fields: "Subject" with the value "I cannot upload video", "Date" with the value "17.05.2021" and a small calendar icon, and "Description" with the value "I cannot upload video with size 2 mb". At the bottom right of this form is a pink button labeled "Send Complaint". To the left of the form, there is a large, semi-transparent teal shape that looks like a stylized 'C'. The entire window has a black border.

Fig 20. Filing a Complaint About a Course

#### 6.1.10 Making an Announcement About a Course

An instructor can make an announcement about his/her course to the enrolled students. In order to make it, he/she needs to press the "**Announce**" button which is next to the course's name on the main page. The button will direct the instructor to the "Make Announcement" page. From there, the instructor should specify a subject, date and a description about the announcements.

Subject  
New lecture coming

Date  
17.05.2021

Description  
I will share a video today, please|

Announce

Fig 21. Making an Announcement About a Course

An instructor can see all of his/her announcements in the Sapientiae system by pressing the “**My Announcements**” button in the navigation bar. From there, the instructor can also delete any announcement by pressing to the “**DELETE**” button.

Course Name	Announcement	
Music Theory for Beginners	We have a discount for this week.	<a href="#">DELETE</a>
SQL 2	New video is not coming this week!	<a href="#">DELETE</a>
Introduction to Living with a Cat	I will share a video today, please go and watch. Good day!	<a href="#">DELETE</a>

Fig 22. My Announcements Page

The screenshot shows a web browser window with the URL [http://localhost:63342/code/src/instructor\\_announcements.php#](http://localhost:63342/code/src/instructor_announcements.php#). The page is titled "Sapientia". The navigation bar includes links for "Account", "Forum", "My Courses", "My Announcements", and "Logout". A red "Add Course" button is located at the top right of the main content area. Below it is a table with two rows. The first row has a "Course Name" column containing "Music Theory for Beginners" and an "Announcement" column containing "We have a discount for this week.", followed by a red "DELETE" button. The second row has a "Course Name" column containing "SQL 2" and an "Announcement" column containing "New video is not coming this week!", followed by another red "DELETE" button.

Course Name	Announcement	
Music Theory for Beginners	We have a discount for this week.	<span style="border: 1px solid red; padding: 2px;">DELETE</span>
SQL 2	New video is not coming this week!	<span style="border: 1px solid red; padding: 2px;">DELETE</span>

Fig 23. My Announcements Page After Deleting the Announcement About the Course  
“Introduction to Living with a Cat”

### 6.1.11 Forum

An instructor can see all the questions from enrolled students about his/her courses from the “**Forum**” button in the navigation bar. From there, the instructor can answer any question about his courses by simply pressing to the “**Answer Question**” button next to the question itself.

The screenshot shows a web-based application interface. At the top, there is a navigation bar with links for 'Account', 'Forum', 'My Courses' (which is underlined in blue), and 'My Announcements'. On the right side of the top bar is a 'Logout' link. Below the navigation bar, the title 'Q&A Section' is centered. A table follows, with a header row containing 'Student Name' and 'Question'. Three rows of data are present in the table:

Student Name	Question
Gamze Guliter 2	I cannot fasten the video, I want 3x.
Gamze Guliter 2	Explain please!
Gamze Guliter 2	asd2

For each question in the table, there is a red-bordered button labeled 'ANSWER QUESTION' on the right side.

Fig 24. Instructor's Forum Page

After pressing the “Answer Question” button, the instructor will be directed to the “Answer the Question” page. From there the instructor can submit an answer for the enrolled student’s question about a course.

The screenshot shows a web-based application interface. At the top, there is a browser-like header with tabs for 'WhatsApp', 'Post Attendee - Zoom', and 'Answer-Instructor'. The URL in the address bar is 'http://localhost:63342/code/src/instructor\_answer.php#'. Below the header, there is a navigation bar with links for 'Account', 'Forum', 'My Courses' (underlined in blue), and 'My Announcements'. On the right side of the top bar is a 'Logout' link. The main content area has a pink header bar with the title 'Answer the Question'. Below the header, the question 'I cannot fasten the video, I want 3x.' is displayed. A text input field with a red border contains the placeholder text 'Write Your Answer Below' and the message 'You cannot do that in Sapientia go use Udemy. :)' followed by a cursor. At the bottom of the input field is a red-bordered 'Post Answer' button. The bottom of the screen features a black footer bar with various icons for 'Mute', 'Start Video', 'Security', 'Participants', 'Chat', 'Share Screen', 'Record', 'Reactions', and a red 'End' button.

Fig 25. Answer the Question Page

### 6.1.12 Statistics

An instructor can see the statistics about his/her courses such as the average student rating, number of enrolled students and the number of refund requests about the course. In order to see the “Statistics” page, the instructor should press to the “Statistics” button which is next to the course’s name in the main page.

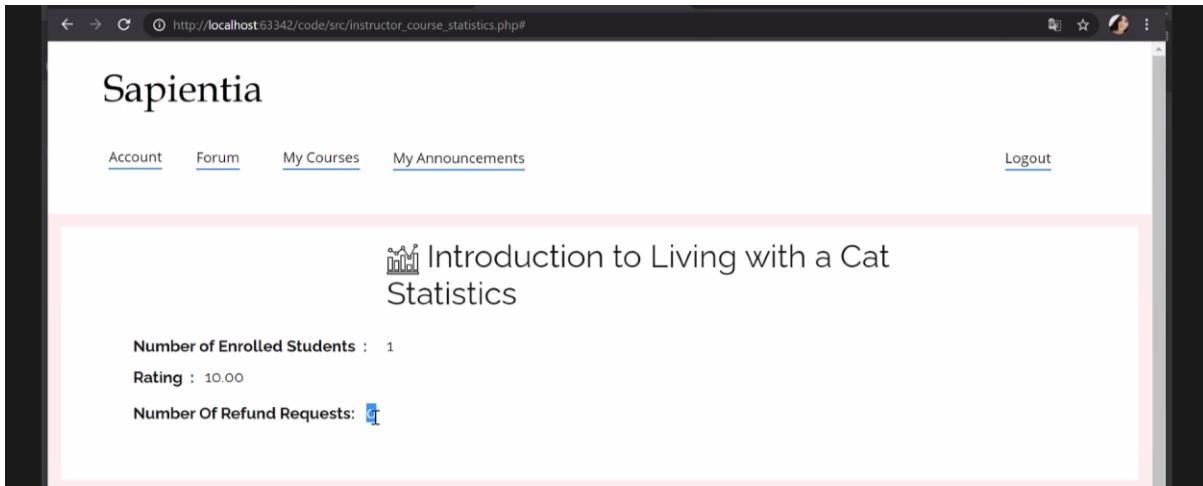


Fig 26. Statistics about the Course “Introduction to Living with a Cat”

## 6.2. Student User Manual

### 6.2.1 Student Sign Up

Students will sign up to the Sapientia website by the Signup Page as entering their name, email, password and clicking the “Sign Up as Student” button.

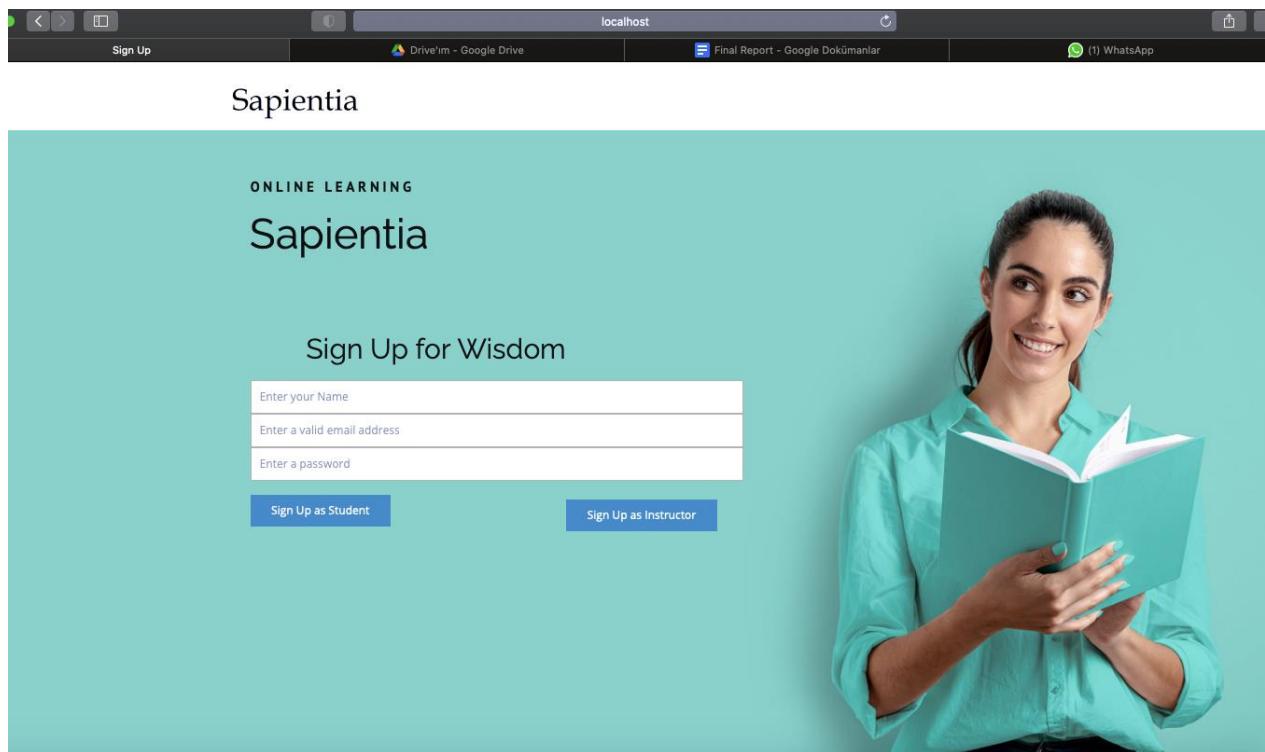


Fig 27. Student Sign Up

### 6.2.2 Student Login

Students will login to the Sapientia website by the Login Page.

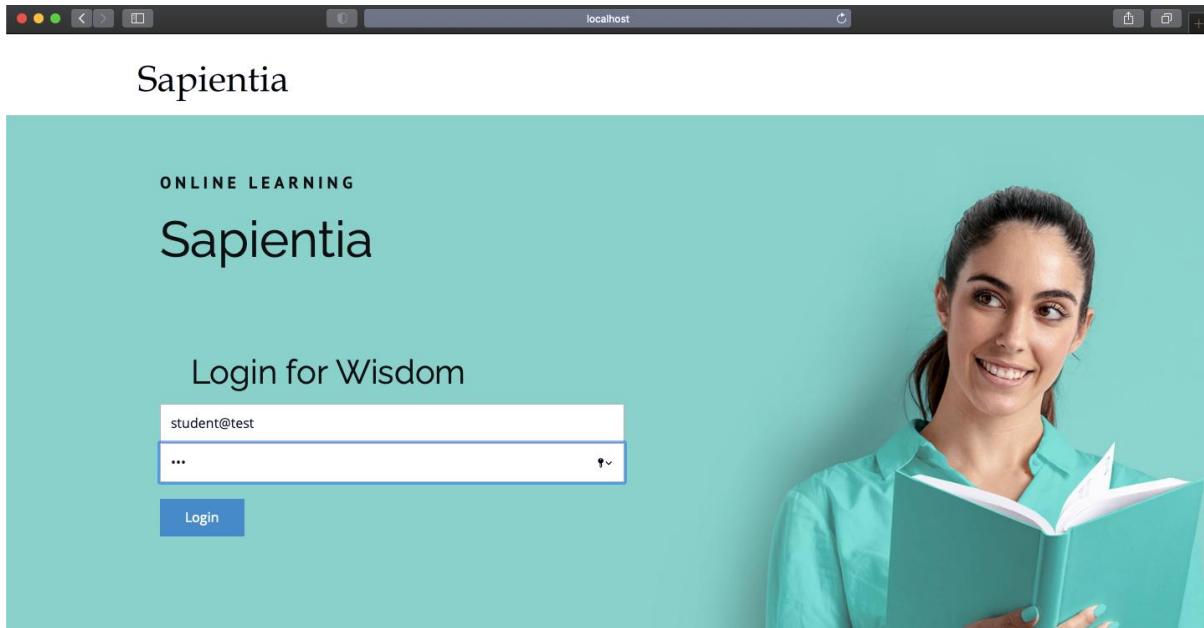


Fig 28. Student Login

### 6.2.3 Student's Main Page

In this page, student will see the courses included in the Sapientia. This page is opened after student logs in. In order to see this page, the student can also use the "**Main Page**" button in the navigation bar. Student can search a course or an instructor by a keyword with the search bar as entering the keyword here and pressing the Enter button on their keyboard. Furthermore, student can filter the course list by course level, category, price and discount by changing the values in the select boxes according to their need and clicking the "**Filter**" button.

Moreover, in the courses table that labeled as "Popular Courses" there are two buttons for each course: "**VIEW**" and "**ADD TO WISHLIST**". By using the "**VIEW**" button, student can see the course page that includes course details such as course description and comments about the course. By using the "**ADD TO WISHLIST**" button, student can add the course to their wishlist.

The figure below shows the page before the filtering.

The figure displays two screenshots of a web application interface for 'Sapientia'.

**Top Screenshot:** This shows the search and filtering interface. It includes a search bar labeled 'Search in Courses or Instructors...', four dropdown menus for filtering by 'Level' (All), 'Category' (All), 'Price' (All), and 'Discount' (All), and a teal-colored 'Filter' button.

**Bottom Screenshot:** This shows the results of the search. The title 'Popular Courses' is displayed above a table. The table has columns: Course Name, Price, Category, Level, Instructor, and two buttons per row: 'VIEW' and 'ADD TO WISHLIST'. The data in the table is as follows:

Course Name	Price	Category	Level	Instructor	VIEW	ADD TO WISHLIST
Music Theory for Beginners	30.99₺ 24.79₺	Music	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
Introduction to Badminton	12.99₺ 10.39₺	Sports	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
SQL 3	45.99₺	Computer Science	Advanced	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
Immanuel Kant	67.99₺	Philosophy	Intermediate	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
Linear Algebra for Beginners	45.89₺	Math	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>

HTML Basics	77.89₺	Graphic Design	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
Introduction to Living with a Cat	30.00₺	Psychology	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
Tulip Science	55.99₺	Gardening	Beginner	Ece Unal	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>

[LEFT](#) [RIGHT](#)

Fig. 29. Student Main Page

After filtering with the keyword ‘sql’ by entering the value to search bar for example:

Popular Courses				
Course Name	Price	Category	Level	Instructor
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin
SQL 3	45.99₺	Computer Science	Advanced	Irem Tekin

[LEFT](#) [RIGHT](#)

Fig. 30. Filtered table by keyword

Listing only advanced level courses:

The screenshot shows a web browser window with the URL `localhost`. The page title is "student main". On the left, there is a sidebar with navigation links: "COURSES", "INSTRUCTORS", "ABOUT", "LOGOUT". The main content area has a search bar at the top with placeholder text "Search in Courses or Instructors...". Below the search bar are several filter dropdowns: "Level" (set to "Advanced"), "Category" (set to "All"), "Price" (set to "All"), and "Discount" (set to "All"). A "Filter" button is located below these dropdowns. The heading "Popular Courses" is displayed above a table. The table has columns: "Course Name", "Price", "Category", "Level", and "Instructor". The data in the table is:

Course Name	Price	Category	Level	Instructor
SQL 3	45.99₺	Computer Science	Advanced	Irem Tekin
Life of Nazim Hikmet	78.99₺	Literature	Advanced	Ece Unal
How to Sell Drugs Online (fast)	199.99₺	Medicine	Advanced	Ece Unal

For each row, there are two buttons: "VIEW" and "ADD TO WISHLIST".

Fig. 31. Filtered table by level

Listing advanced courses in Computer Science category:

This screenshot is similar to Fig. 31, showing the same interface but with different filter settings. The "Category" dropdown is now set to "Computer Science". The rest of the filters and the table content are identical to Fig. 31.

The table data is:

Course Name	Price	Category	Level	Instructor
SQL 3	45.99₺	Computer Science	Advanced	Irem Tekin

For the single row, there are "VIEW" and "ADD TO WISHLIST" buttons. At the bottom of the table, there are "LEFT" and "RIGHT" navigation buttons. The page number "1" is also visible.

Fig. 32. Filtered table by label and category

Listing courses between prices 35-50 liras:

Level  
All

Category  
All

Price  
\$15-\$35

Discount  
All

Filter

### Popular Courses

Course Name	Price	Category	Level	Instructor		
Music Theory for Beginners	30.99₺ 24.79₺	Music	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
Introduction to Living with a Cat	30.00₺	Psychology	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>

Fig. 33. Filtered table by price

Listing courses between prices 35-50 liras and includes “sql” keyword:

Level  
All

Category  
All

Price  
\$15-\$35

Discount  
All

Filter

### Popular Courses

Course Name	Price	Category	Level	Instructor		
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>

[LEFT](#) [RIGHT](#)

Fig. 34. Filtered table by price and keyword

There are 10 courses displayed per table page. To see the remaining courses, user can click the left and right buttons to navigate to table page. In the screen shot of the website below, the second page of the table is shown since user clicked the “RIGHT” button to increment the page number.

The screenshot shows a web browser window titled "localhost". At the top, there are two dropdown menus: one for "All" and another for "Discount", both currently set to "All". Below them is a "Filter" button. The main content area is titled "Popular Courses" and displays a table with the following data:

Course Name	Price	Category	Level	Instructor		
Life of Nazim Hikmet	78.99₺	Literature	Advanced	Ece Unal	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
How to Sell Drugs Online (fast)	199.99₺	Medicine	Advanced	Ece Unal	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>
CS 353	80.00₺	Gardening	Intermediate	Mustafa Cavdar2	<a href="#">VIEW</a>	<a href="#">ADD TO WISHLIST</a>

At the bottom of the table, there are "LEFT" and "RIGHT" navigation buttons. A small number "2" is centered below the buttons, indicating this is the second page of results.

Fig. 35. Second page of table

#### 6.2.4 Course Page

After clicking the “VIEW” button at main page, student can see the course instructor(s), price, rating, category, level for the selected course. Furthermore, at the bottom part of the page, student can see the comments for the course and at the right side, the description of the course is shown. Before purchasing the course, there are two buttons displayed in this page: “ADD TO WISHLIST” and “BUY”. As the names suggest, student can add the course to their wishlist or buy the course by clicking those

buttons.

The screenshot shows a web browser window with the address bar displaying 'localhost'. The main content area is titled 'Sapientia'. Below the title, there is a navigation bar with links: 'View Course', 'Account', 'Notifications', 'My Courses' (which is underlined), 'Fill a Complaint', 'Certificates', 'Wishlist', 'Main Page', and 'Logout'. The main content area features a large teal-bordered box containing course details for 'SQL 3'. Inside this box, there are two columns of information:

Instructor(s):	Irem Tekin
Price :	45.99₺
Rating :	Not rated yet!
Category:	Computer Science
Level:	Advanced

On the right side of the box, there is a section titled 'About the Course:' with the text 'SQL 3 for advance users.' Below the course details, there are two red rectangular buttons: 'Add to Wishlist' and 'Buy'. At the bottom left of the teal box, there is a link 'Course Comments' followed by a table with two columns: 'Graduated Student Name' and 'Comment'.

Fig. 36. Course page before purchase

### 6.2.5 After Buying the Course - Course Page

After purchasing the course, course page changes to the version shown below. Student can reach this page from the list that will be displayed by clicking the “**My Courses**” at the navigation bar.

By clicking the “**View Lectures**” button, student can be navigated to the page that shows course contents, which are lectures and quizzes.

The screenshot shows a web browser window with three tabs: 'Course Page' (active), 'localhost' (Google Drive), and 'Final Report - Google Dokümanlar'. The main content area is titled 'SQL 3' and displays course details: Instructor(s) Irem Tekin, Rating Not rated yet!, Category Computer Science, Level Advanced. It also shows 'About the Course: SQL 3 for advance users.', 'Completeness: 0%', and a 'View Lectures' button. Below this is a 'Ask a Question' form with fields for 'Subject' (containing 'Question Subject') and 'Question' (containing 'Question'). A 'Post Question' button is at the bottom. To the right, there's a link 'To see all Q&A's: Go to Course Forum'.

Fig. 37. Course page after purchase

by filling the form that labeled as “**Ask a Question**” student can ask a question about the course to the instructor. In this form, it is asked to write the subject of the question and question itself. After that, student can publish the question by clicking the “**Post Question**” button.

The image shows a close-up of the 'Ask a Question' form. It has two input fields: 'Subject' containing 'Q1?' and 'Question' containing 'Question question'. A large blue border surrounds the 'Question' field. At the bottom is a red 'Post Question' button.

With the help of the “**Go to Course Forum**” student can go to the forum page where the Q&As are displayed.

To see all Q&A's : [Go to Course Forum](#)

Fig. 38. and 39. Ask Question and Go to Course Forum

#### **6.2.6 Course Forum**

In this page, student can view all of the questions that asked to the instructor. Also, by clicking the “**VIEW ANSWER**” button, student will be navigated to the answer page for the related question.

The screenshot shows a web browser window with the title 'localhost'. The tabs at the top include 'Forum', 'Drive' (with 'Drive' and 'Final Report'), and 'WhatsApp'. The main content area is titled 'Sapientia' and has a navigation bar with links for 'Account', 'Notifications', 'My Courses', 'Fill a Complaint', 'Certificates', 'Wishlist', 'Main Page', and 'Logout'. Below this is a teal header bar with the text 'Q&A Section'. A table lists four student questions, each with a 'VIEW ANSWER' button.

Student Name	Question	
Gamze Gulter 2	I cannot fasten the video, I want 3x.	<a href="#">VIEW ANSWER</a>
Gamze Gulter 2	Explain please!	<a href="#">VIEW ANSWER</a>
Gamze Gulter 2	asd2	<a href="#">VIEW ANSWER</a>
Student Test	Question question	<a href="#">VIEW ANSWER</a>

Fig. 40. Course forum

#### 6.2.7 View answer to a forum question

Student can see the answer of the question that they wanted to learn in this page.

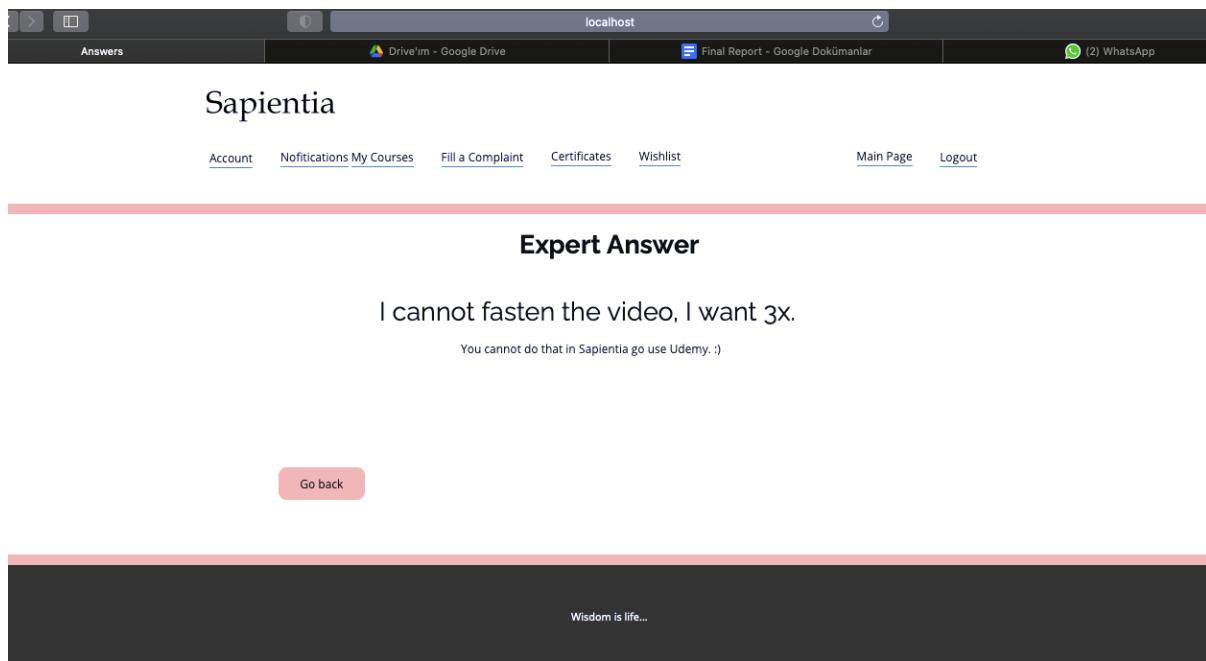


Fig. 41. Answer of the question

#### 6.2.8 See Contents of the Course

Student can see the contents of a course, which are lectures and quizzes in this page. **We divided contents as sections to achieve a more neat look.**

The screenshot shows a web browser window with the address bar displaying 'localhost'. The title bar says 'Sapientia'. The navigation menu includes 'Contents', 'Account', 'Notifications', 'My Courses', 'Fill a Complaint', 'Certificates', 'Wishlist', 'Main Page', and 'Logout'. Below the menu, a pink horizontal bar contains the word 'Contents'. The main content area has a light gray background. It features a table with three sections: 'Section 1' (Fundamentals), 'Section 2' (What is Music?), and 'Section 3' (Grand Staff). 'Section 1' contains three rows: 'Lecture: Welcome to the Music Theory' with a 'View Lecture' button, 'Lecture: Rhythm Theory' with a 'View Lecture' button, and 'Quiz: Rhythm Quiz 1' with a 'Solve Quiz' button. 'Section 2' and 'Section 3' each have a single row with their respective titles. At the bottom of the page is a dark gray footer bar with the text 'Wisdom is life...'.

Fig. 42. Contents of a course

### 6.2.9 Watch Lecture

In this page, student can watch the lecture video and take notes. Student can stop the video by clicking the video image. Also, by clicking the “**Back to content**” button, user can navigate to the Contents page. With clicking “**Previous**” and “**Next**” buttons, student can go to previous and next lecture pages, respectively.

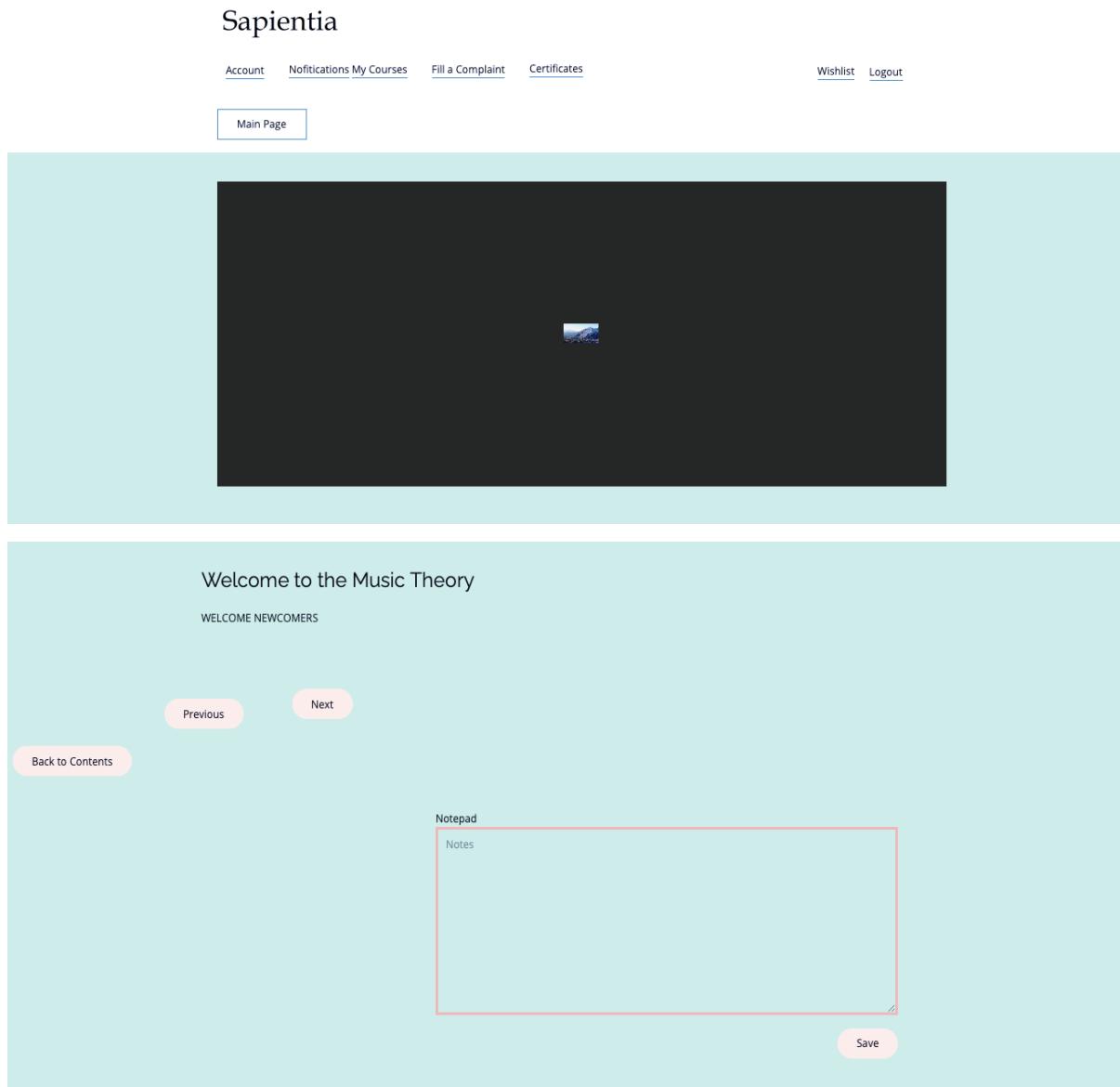


Fig. 43. Watch Lecture page

By filling the text area shown below and hitting “Save” button, student can take notes for the related lecture.

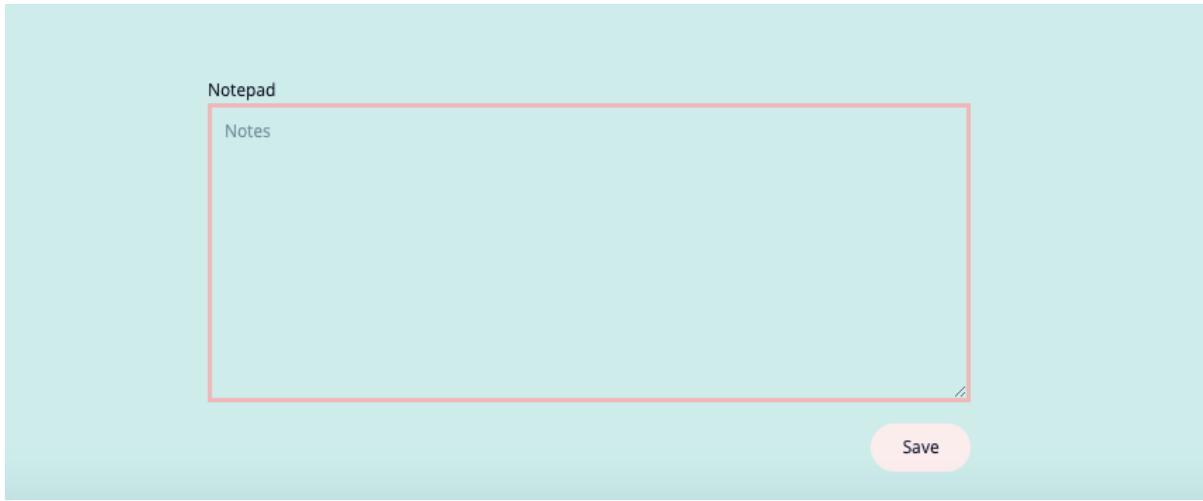


Fig. 44. Notepad section

#### 6.2.10 Solve Quiz

In this page, student can see the quiz question and select one of the choices. With clicking “Previous” and “Next” buttons, student can go to previous and next lecture quiz questions, respectively. Finally, student can submit all of their answers for related quiz by clicking the “Submit Quiz” button. Furthermore, student can see their grade after submitting the quiz, and by clicking the “Back to Contents” button, they can navigate to the Contents page.

# Music Theory for Beginners

## Rhythm Quiz 1

Question

What is rhythm?

smtg about computers (maybe)  
 a strong, regular repeated pattern of movement or  
 no idea!

[Previous](#) [Next](#) [Submit Quiz](#)

Grade : .../ 10

[Back to Contents](#)

Fig. 45. Solve Quiz Page

### 6.2.11 View Wishlist

Student can see their wishlist in this page. By clicking the “VIEW”, “BUY”, “REMOVE” buttons, student can view the course page, buy the course or remove the course from the wishlist respectively.

The screenshot shows a web browser window titled 'localhost' with several tabs open: 'Wishlist', 'Drive'im - Google Drive', 'Final Report - Google Dokümanlar', and '(2) WhatsApp'. The main content area is titled 'Sapientia' and has a navigation bar with links: Account, Notifications, My Courses, Fill a Complaint, Certificates, Wishlist, Main Page, and Logout. Below this is a section titled 'Wishlist' containing a table with one row. The table columns are Course Name, Price, Category, Level, and Instructor. The row data is: SQL 3, 45.99₺, Computer Science, Advanced, Irem Tekin. To the right of the table are three buttons: VIEW, BUY, and REMOVE. Below the table are two small buttons labeled 'LEFT' and 'RIGHT'. At the bottom center is the number '1'. A dark banner at the bottom of the page contains the text 'Wisdom is life...'.

Fig. 46. Wishlist page

### 6.2.12 View Certificates

In this page, student can see the list of certificates that they got. By clicking the “VIEW CERTIFICATE” as the name indicates, student can see the certificate for the related course.

The screenshot shows a web browser window titled 'localhost' with tabs: 'certificates', 'Drive'im - Google Drive', 'Final Report - Google Dokümanlar', and '(2) WhatsApp'. The main content area is titled 'Sapientia' and has a navigation bar with links: Account, Notifications, My Courses, Fill a Complaint, Certificates, Wishlist, Main Page, and Logout. Below this is a section titled 'My Certificates' containing a table with two rows. The table column is Course Name. The rows are: Music Theory for Beginners and Introduction to Living with a Cat. To the right of each row is a 'VIEW CERTIFICATE' button. A dark banner at the bottom of the page contains the text 'Wisdom is life...'.

Fig. 47. View Certificates page

### 6.2.13 Certificate Page

Student can see the certificate for a course that they successfully completed in this page. Moreover, student can send comment about the course by filling the text area labeled as “**Comment Course**” and clicking the “**Send Comment**” button. Also student can rate the course by filling the text area labeled as “**Rate Course**” and clicking the “**Rate**” button.

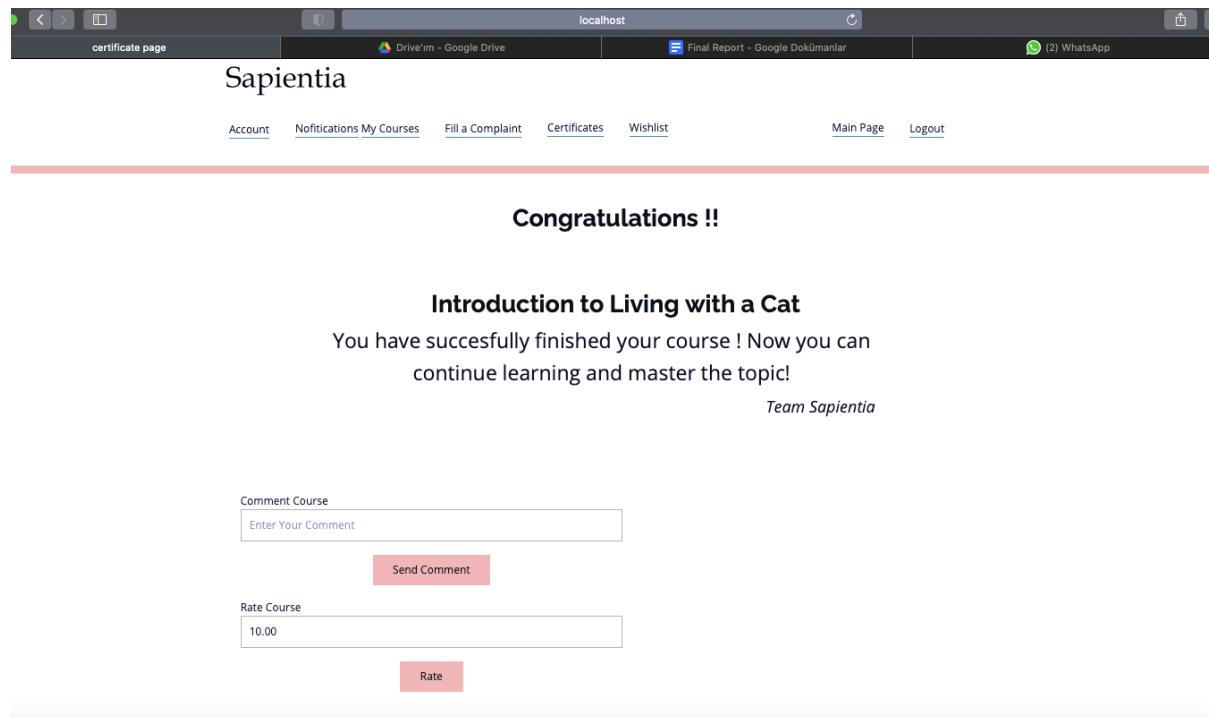


Fig. 48. Certificate page

### 6.2.14 Filling a Complaint About a Course

A student can add a complaint about a course by the help of this page. In order to fill a complaint he/she needs to press to the “**Fill a Complaint**” button which is next to the course’s name in the main page. From this page, the student must specify a subject, date and a brief

description about the complaint and press to the “**Send Complaint**” button. Also, selecting the “**Refund Request**” choice, student can request a refund from the administration.

The screenshot shows a web browser window titled "localhost" with the URL "Fill a complaint - student". The page is part of the Sapientia platform, as indicated by the logo at the top. The main content is titled "Fill a Complaint". A red rectangular box highlights the form fields for "Subject", "Date", and "Description". Below these fields are two radio buttons: "Refund Request" and "Other". A pink rounded rectangle button labeled "Send Complaint" is located at the bottom right of the highlighted area. The background features a large teal abstract shape.

Fig. 49. Fill a Complaint page

## 6.3. Admin User Manual

### 6.3.1 Admin Login

Sapientia admin can login to the system using the login page.

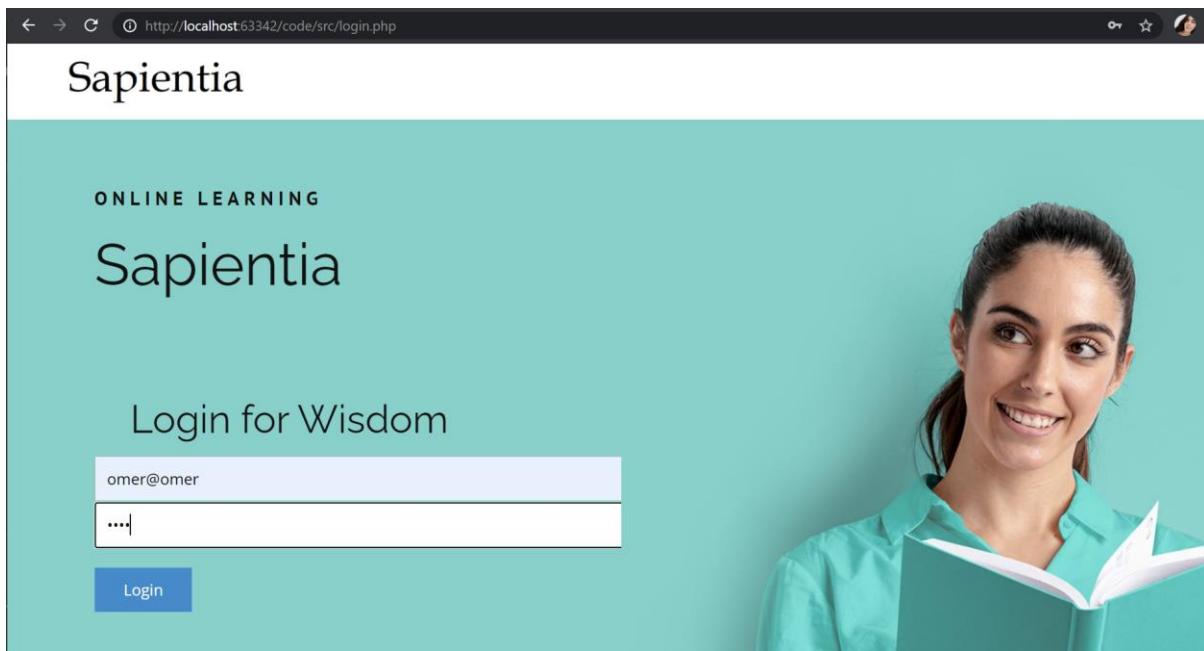


Fig. 49. Admin Login

A screenshot of a database management interface showing an "Admin" table. The table has columns: AID, name, e\_mail, and password. There is one row of data: AID is 1, name is omer, e\_mail is omer@omer, and password is 1234. The interface includes a toolbar with various icons and a "Filter Criteria" section.

Fig. 50. Admin Table

### 6.3.2 Admin Main Page and Filtering

Sapientia admin sees a main page which consists of all courses that are added by instructors. Admin can filter the courses, view the courses statistics and offer discount using this page.

The screenshot shows a web browser window with the URL [http://localhost:63342/code/src/admin\\_main.php#](http://localhost:63342/code/src/admin_main.php#). At the top, there is a search bar labeled "Search in Courses or Instructors...". To the right of the search bar are five dropdown menus for filtering: "Level" (Advanced), "Category" (All), "Price" (All), "Discount" (All), and a "Filter" button. Below the filtering section, the word "Courses" is centered above a table. The table has a header row with columns: Course Name, Price, Category, Level, and Instructor. The data row for "SQL 3" shows: Course Name (SQL 3), Price (45.99₺ 29.43₺), Category (Computer Science), Level (Advanced), and Instructor (Irem Tekin). To the right of the "Instructor" column are two buttons: "VIEW STATISTICS" and "OFFER DISCOUNT".

Course Name	Price	Category	Level	Instructor
SQL 3	45.99₺ 29.43₺	Computer Science	Advanced	Irem Tekin

Courses

[VIEW STATISTICS](#)   [OFFER DISCOUNT](#)

Fig. 51. Admin Main Page and Filtering

### 6.3.3 View Course Statistics

Admin can view the statistics about a course by clicking the **view statistics button**. In statistics page, the number of students who enrolled the course, the rating of the course and the number of refund requests for the course are displayed.

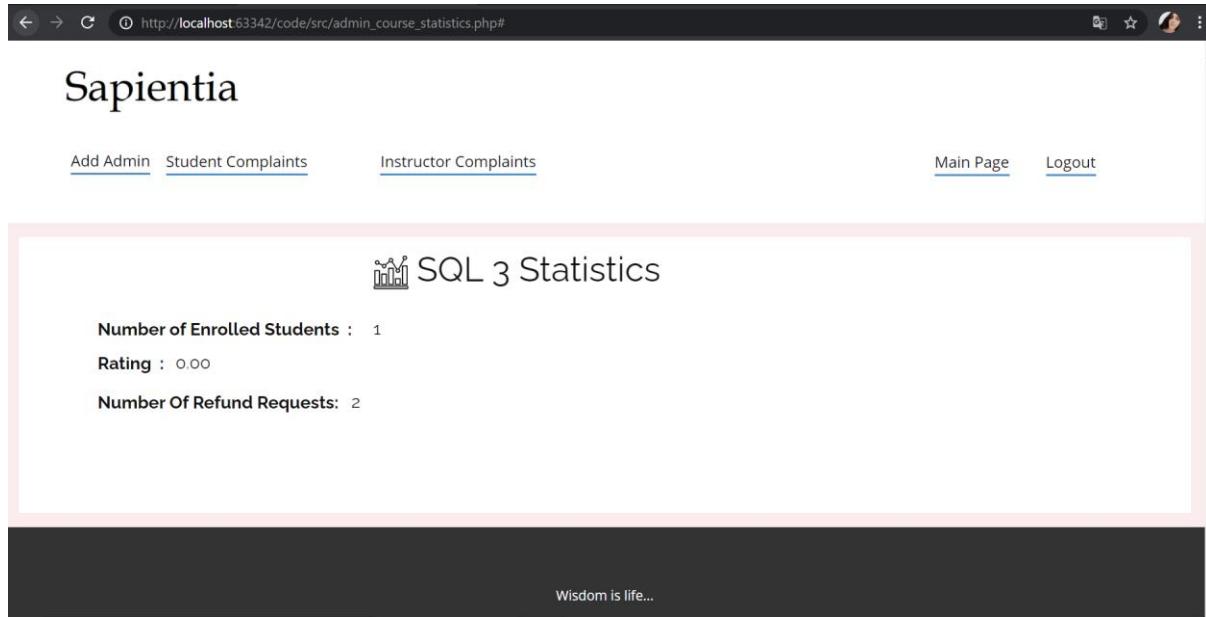


Fig. 52. View Course Statistics

#### 6.3.4 Discount Management - Apply Discount

In the main page, the **offer discount button** is displayed for the courses that are allowed to get discounts. Admin can manage discounts of a course by clicking the **offer discount button**. The button leads admin to course's price and discount information. Admin can change the discount rate or delete the discount. To apply discount, admin writes a new discount rate, selects apply discount and saves changes. The change in the price of the course is displayed on the main page.

[http://localhost:63342/code/src/admin\\_main.php#](http://localhost:63342/code/src/admin_main.php#)

## Courses

Course Name	Price	Category	Level	Instructor	
Music Theory for Beginners	30.99₺ 24.79₺	Music	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>
Introduction to Badminton	12.99₺ 10.39₺	Sports	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a> <a href="#">OFFER DISCOUNT</a>
SQL 3	45.99₺ 29.43₺	Computer Science	Advanced	Irem Tekin	<a href="#">VIEW STATISTICS</a> <a href="#">OFFER DISCOUNT</a>
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin	<a href="#">VIEW STATISTICS</a>

[http://localhost:63342/code/src/admin\\_offer\\_discount.php#](http://localhost:63342/code/src/admin_offer_discount.php#)

## Sapientia

Add Admin [Student Complaints](#) [Instructor Complaints](#) Main Page Logout

### SQL 3

Determined Price by Instructor (Discount will be applied to this price)

45.99

Applied Discount Rate

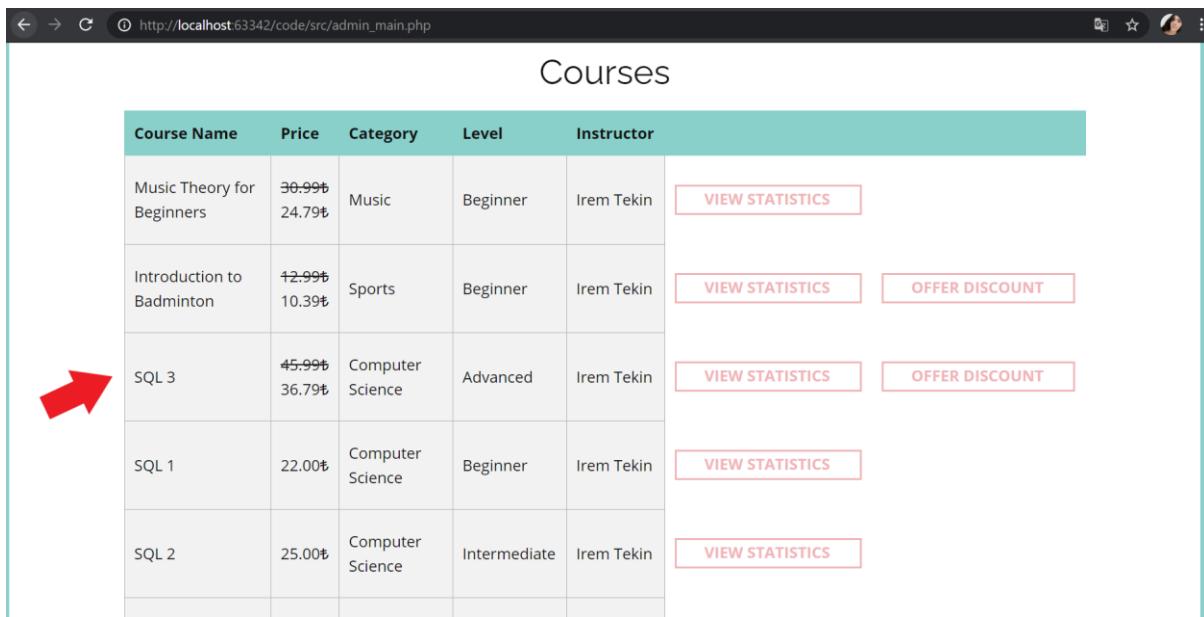
36.00

New Discount Rate

20

Apply Discount  
 Delete Discount

[Save Changes](#)

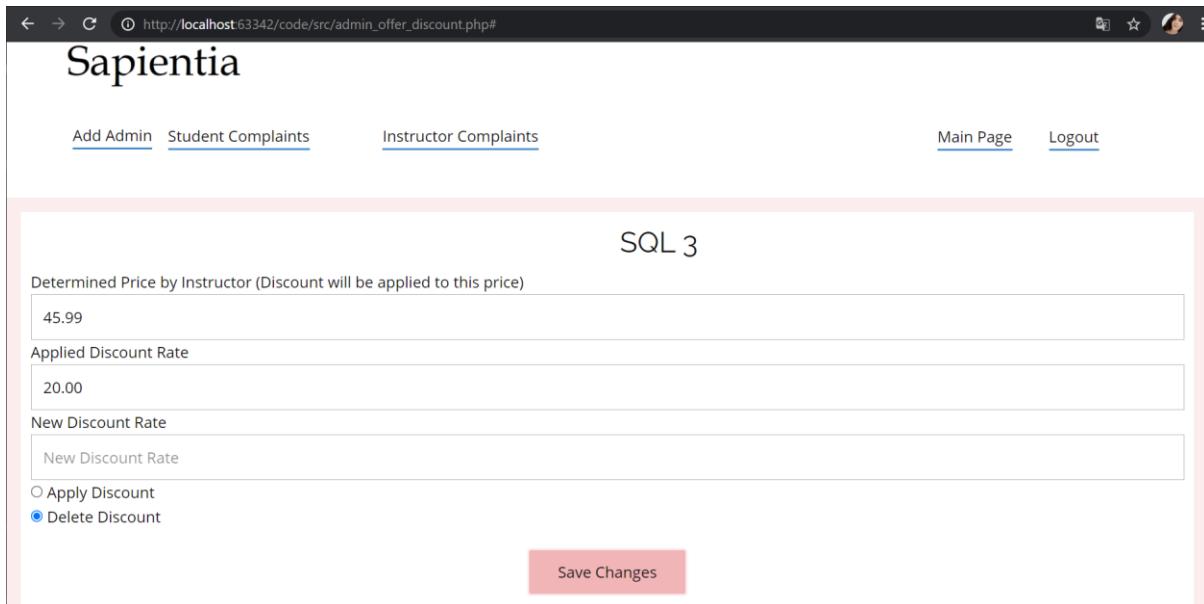


Course Name	Price	Category	Level	Instructor	
Music Theory for Beginners	30.99₺ 24.79₺	Music	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>
Introduction to Badminton	12.99₺ 10.39₺	Sports	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a> <a href="#">OFFER DISCOUNT</a>
SQL 3	45.99₺ 36.79₺	Computer Science	Advanced	Irem Tekin	<a href="#">VIEW STATISTICS</a> <a href="#">OFFER DISCOUNT</a>
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin	<a href="#">VIEW STATISTICS</a>

Fig. 53., 54. and 55. Apply Discount

#### 6.3.4 Discount Management - Delete Discount

Admin can also delete an already applied discount by selecting delete discount and select changes. The change in the price of the course is displayed on the main page.



Determined Price by Instructor (Discount will be applied to this price)  
45.99

Applied Discount Rate  
20.00

New Discount Rate  
New Discount Rate

Apply Discount  
 Delete Discount

[Save Changes](#)



Introduction to Badminton	12.99₺ 10.39₺	Sports	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>	<a href="#">OFFER DISCOUNT</a>
SQL 1	22.00₺	Computer Science	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>	
SQL 2	25.00₺	Computer Science	Intermediate	Irem Tekin	<a href="#">VIEW STATISTICS</a>	
SQL 3	45.99₺	Computer Science	Advanced	Irem Tekin	<a href="#">VIEW STATISTICS</a>	<a href="#">OFFER DISCOUNT</a>
Immanuel Kant	67.99₺	Philosophy	Intermediate	Irem Tekin	<a href="#">VIEW STATISTICS</a>	
Linear Algebra for Beginners	45.89₺	Math	Beginner	Irem Tekin	<a href="#">VIEW STATISTICS</a>	<a href="#">OFFER DISCOUNT</a>

Fig. 56. and 57. Delete Discount

### 6.3.5 Manage Student Complaints

Students can complain about courses, in their complaints can ask for a refund or not. Admin can view all the complaints by clicking **Student Complaints button** in the header. If a student asks for a refund, **refund button** is displayed next to the student complaint. If student doesn't ask for a refund, **noted button** is displayed next to **ignore button**.

#### 6.3.5.1 Ignore Student Complaint

Admin can ignore a complaint by clicking **ignore button**. When admin ignores the complaint the complaint is deleted.

http://localhost:63342/code/src/admin\_student\_complaints.php

Course Name	Student Name	Complaint Subject	Complaint	Date	
Music Theory for Beginners	Gamze Guliter 2	Course could not satisfy my ex	Apparently I have some other type of expectations from a course (you know ;)	2021-05-17	<a href="#">IGNORE</a> <a href="#">REFUND</a>
SQL 3	Gamze Guliter 2	I hate this course	I hate it	2021-05-17	<a href="#">IGNORE</a> <a href="#">REFUND</a>
SQL 3	Gamze Guliter 2	Video Fast Mode Problem	123	2021-05-17	<a href="#">IGNORE</a> <a href="#">NOTED</a>
SQL 3	Gamze Guliter 2	complaint1	wd	2021-05-17	<a href="#">IGNORE</a> <a href="#">NOTED</a>

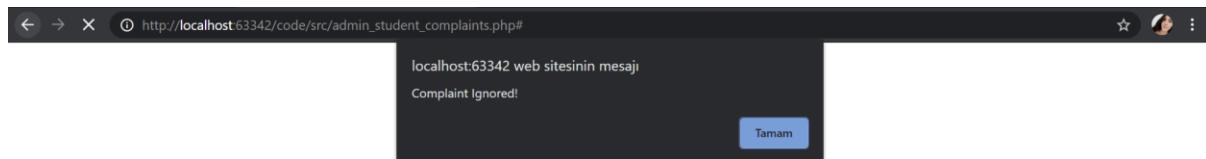


Fig. 58. and 59. Ignore Student Complaint

http://localhost:63342/code/src/admin\_student\_complaints.php#

Course Name	Student Name	Complaint Subject	Complaint	Date	
SQL 3	Gamze Guliter 2	i hate this course	i hate it	2021-05-17	<a href="#">IGNORE</a> <a href="#">REFUND</a>
SQL 3	Gamze Guliter 2	Video Fast Mode Problem	123	2021-05-17	<a href="#">IGNORE</a> <a href="#">NOTED</a>
SQL 3	Gamze Guliter 2	complaint1	wd	2021-05-17	<a href="#">IGNORE</a> <a href="#">NOTED</a>
SQL 3	Gamze Guliter 2	sdfg	qwefg	2021-05-17	<a href="#">IGNORE</a> <a href="#">REFUND</a>

Fig. 60. After ignore Student Complaint

### 6.3.5.2 Note Student Complaint

Admin can use **noted button** if admin doesn't want to ignore a complaint which doesn't include a refund request.

http://localhost:63342/code/src/admin\_student\_complaints.php#

Course Name	Student Name	Complaint Subject	Complaint	Date	
SQL 3	Gamze Guliter 2	i hate this course	i hate it	2021-05-17	<a href="#">IGNORE</a> <a href="#">REFUND</a>
SQL 3	Gamze Guliter 2	Video Fast Mode Problem	123	2021-05-17	<a href="#">IGNORE</a> <a href="#">NOTED</a>
SQL 3	Gamze Guliter 2	complaint1	wd	2021-05-17	<a href="#">IGNORE</a> <a href="#">NOTED</a>
SQL 3	Gamze Guliter 2	sdfg	qwefg	2021-05-17	<a href="#">IGNORE</a> <a href="#">REFUND</a>

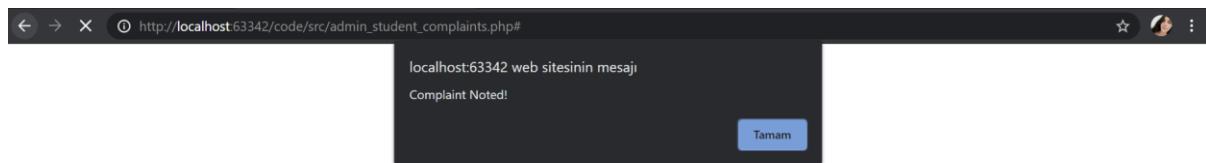


Fig. 61. and 62. Note Student Complaint

A screenshot of the Sapientia application. The title bar says "Sapientia". The navigation menu includes "Add Admin", "Student Complaints" (which is underlined), "Instructor Complaints", "Main Page", and "Logout". The main content area is titled "Student Complaints" and contains a table with three rows of student complaints. The table has columns: Course Name, Student Name, Complaint Subject, Complaint, and Date. The first row has complaints "i hate this course" and "i hate it" with dates "2021-05-17" and buttons "IGNORE" and "REFUND". The second row has complaints "complaint1" and "wd" with date "2021-05-17" and buttons "IGNORE" and "NOTED". The third row has complaints "sdfg" and "qwefg" with date "2021-05-17" and buttons "IGNORE" and "REFUND".

Course Name	Student Name	Complaint Subject	Complaint	Date		
SQL 3	Gamze Guliter 2	i hate this course	i hate it	2021-05-17	<a href="#">IGNORE</a>	<a href="#">REFUND</a>
SQL 3	Gamze Guliter 2	complaint1	wd	2021-05-17	<a href="#">IGNORE</a>	<a href="#">NOTED</a>
SQL 3	Gamze Guliter 2	sdfg	qwefg	2021-05-17	<a href="#">IGNORE</a>	<a href="#">REFUND</a>

Fig. 63. After Note Student Complaint

### 6.3.5.3 Refund Student

http://localhost:63342/code/src/admin\_student\_complaints.php

Add Admin Student Complaints Instructor Complaints Main Page Logout

### Student Complaints

Course Name	Student Name	Complaint Subject	Complaint	Date	
SQL 3	Gamze Guliter 2	i hate this course	i hate it	2021-05-17	<button>IGNORE</button> <button>REFUND</button>
SQL 3	Gamze Guliter 2	complaint1	wd	2021-05-17	<button>IGNORE</button> <button>NOTED</button>
SQL 3	Gamze Guliter 2	sdfg	qwefg	2021-05-17	<button>IGNORE</button> <button>REFUND</button>

http://localhost:63342/code/src/admin\_student\_complaints.php#

localhost:63342 web sitesinin mesajı  
Complaint accepted, Refund Approved!

Tamam

Fig. 64. and 65. Refund Student

http://localhost:63342/code/src/admin\_student\_complaints.php#

Add Admin Student Complaints Instructor Complaints Main Page Logout

### Student Complaints

Course Name	Student Name	Complaint Subject	Complaint	Date	
SQL 3	Gamze Guliter 2	complaint1	wd	2021-05-17	<button>IGNORE</button> <button>NOTED</button>
SQL 3	Gamze Guliter 2	sdfg	qwefg	2021-05-17	<button>IGNORE</button> <button>REFUND</button>

Fig. 66. After Refund

### 6.3.6 Manage Instructor Complaints

Instructors can complain about their courses to admin. Admin can ignore or note the complaints.

The screenshot shows a web browser window with the URL [http://localhost:63342/code/src/admin\\_instructor\\_complaint.php](http://localhost:63342/code/src/admin_instructor_complaint.php). The page title is "Sapientia". The navigation menu includes "Add Admin", "Student Complaints", "Instructor Complaints", "Main Page", and "Logout". The main content area is titled "Instructor Complaints" and contains a table with two rows of data. The columns are "Course Name", "Instructor Name", "Complaint Subject", "Complaint", and "Date". The first row has a complaint about course discount and the second about video upload size. To the right of each row are two buttons: "IGNORE" and "NOTED". Red arrows point from the text "Fig. 67. Ignore and Note the Complaints" to these buttons. Below the table are "LEFT" and "RIGHT" navigation buttons.

Course Name	Instructor Name	Complaint Subject	Complaint	Date	
Music Theory for Beginners	Irem Tekin	Discount	How can i open my course for discount allow?	2021-05-17	<input type="button" value="IGNORE"/> <input type="button" value="NOTED"/>
Introduction to Living with a Cat	Irem Tekin	I cannot upload video	I cannot upload video with size 2 mb	2021-05-17	<input type="button" value="IGNORE"/> <input type="button" value="NOTED"/>

Fig. 67. Ignore and Note the Complaints

The screenshot shows the same web browser window and page structure as Fig. 67. However, the "IGNORE" and "NOTED" buttons have been clicked for both complaints in Fig. 67. The table now shows the complaints without these buttons. Red arrows point from the text "Fig. 68. After Ignore and Note the Complaints" to the empty button areas where the buttons were previously located.

Course Name	Instructor Name	Complaint Subject	Complaint	Date	
Music Theory for Beginners	Irem Tekin	Discount	How can i open my course for discount allow?	2021-05-17	
Introduction to Living with a Cat	Irem Tekin	I cannot upload video	I cannot upload video with size 2 mb	2021-05-17	

Fig. 68. After Ignore and Note the Complaints

### 6.3.7 Add Admin

An admin can add other admins by clicking the **add admin button** in the header and giving information about a new admin.

ONLINE LEARNING

# Sapientia

## Add Admin

Merve
merve@merve
...

Add admin

Fig. 69. Add Admin

	AID	name	e_mail	password
1	1	omer	omer@omer	1234
2	3	Merve	merve@merve	123

Fig. 70. Admin Table

## **7. Website**

The link of GitHub page of the project is the following:

[https://github.com/Sapientia-Course-Platform/CS353\\_Project](https://github.com/Sapientia-Course-Platform/CS353_Project)

The link of Reports Website is the following:

[https://sapientia-course-platform.github.io/CS353\\_Project/](https://sapientia-course-platform.github.io/CS353_Project/)

## 9. Appendix

### 8.1. SQL Create Table Code

```
use omer_unlusoy;

DROP TABLE IF EXISTS Take_Quiz_Question;
DROP TABLE IF EXISTS Take_Lecture;
DROP TABLE IF EXISTS Take_Quiz;
DROP TABLE IF EXISTS Zoom_Session;
DROP TABLE IF EXISTS Discount;
DROP TABLE IF EXISTS Evaluates_Complaint_Entry_Student;
DROP TABLE IF EXISTS Evaluates_Complaint_Entry_Instructor;
DROP TABLE IF EXISTS Teaches;
DROP TABLE IF EXISTS Enrolls;
DROP TABLE IF EXISTS Wishlist;
DROP TABLE IF EXISTS Certificate;
DROP TABLE IF EXISTS QnA_Entry_Student;
DROP TABLE IF EXISTS QnA_Entry_Instructor;
DROP TABLE IF EXISTS Complaint_Entry_Student;
DROP TABLE IF EXISTS Complaint_Entry_Instructor;
DROP TABLE IF EXISTS Course_Announcement;
DROP TABLE IF EXISTS Note;
DROP TABLE IF EXISTS Quiz_Question;
DROP TABLE IF EXISTS Quiz;
DROP TABLE IF EXISTS Lecture;
DROP TABLE IF EXISTS Section;
DROP TABLE IF EXISTS Course;

DROP TABLE IF EXISTS Student;
DROP TABLE IF EXISTS Instructor;
DROP TABLE IF EXISTS Admin;

/*
General Assumptions:
ID          -> INT AUTO_INCREMENT
```

```

referencing ID      -> INT NOT NULL
short string        -> VARCHAR(30)
mid string          -> VARCHAR(50)
long string         -> VARCHAR(70)
description         -> VARCHAR(300)
money               -> NUMERIC(12, 2) OR NUMERIC(8, 2)
%                   -> NUMERIC(5, 2)
date                -> DATE
URL                -> VARCHAR(1024)

Special Cases:
membership_type    -> ENUM('BRZ', 'SLV', 'GLD')
refund_request     -> BOOLEAN
duration            -> TIME
lecture_content     -> BLOB NOT NULL
membership_type     -> ENUM('BRZ', 'SLV', 'GLD') NOT NULL DEFAULT 'BRZ'
answer              -> ENUM('choice1', 'choice2', 'choice3') NOT NULL DEFAULT
'choice1'

Constraints:
Student-Course cannot be in Wishlist, Enrolls, and Certificate all together.
adds Wishlist -> Enrolls (remove from Wishlist) -> get Certificate (add
Certificate not remove Enrolls)
Lecture and Quiz content_num

Future:
ENGINE = InnoDB; can be added

*/
/* Assumptions:
*/
CREATE TABLE Student(
    SID           INT AUTO_INCREMENT,
    name          VARCHAR(50) NOT NULL,
    e_mail        VARCHAR(50) NOT NULL UNIQUE,

```

```

        password      VARCHAR(30) NOT NULL,
        membership_type ENUM('BRZ', 'SLV', 'GLD') NOT NULL
DEFAULT 'BRZ',
        PRIMARY KEY (SID)
);

/* Assumptions:
   include functions [course_count(), current_income()]
*/
CREATE TABLE Instructor(
        IID          INT AUTO_INCREMENT,
        name         VARCHAR(50) NOT NULL,
        e_mail       VARCHAR(50) NOT NULL UNIQUE,
        password     VARCHAR(30) NOT NULL,
        PRIMARY KEY (IID)
);

CREATE TABLE Admin(
        AID          INT AUTO_INCREMENT,
        name         VARCHAR(50) NOT NULL,
        e_mail       VARCHAR(50) NOT NULL UNIQUE,
        password     VARCHAR(30) NOT NULL,
        salary       NUMERIC(12, 2) NOT NULL DEFAULT 0.00,
        PRIMARY KEY (AID)
);

/* Assumptions:
   include functions [rating(), student_count(), lecture_count()]
constraints for category and level
0 <= cost <= 999,999.99
discount_allowed = True OR False
lecture_completed_threshold = %70
0 <= certificate_threshold <= 100.00
attention: FOREIGN KEY (creator_IID) REFERENCES Instructor(IID) ON DELETE
NO ACTION ON UPDATE CASCADE
        UNIQUE (creator_IID, course_name)
*/
CREATE TABLE Course(

```

```

        CID           INT AUTO_INCREMENT,
        IID           INT NOT NULL,
        course_name   VARCHAR(70) NOT NULL,
        description   VARCHAR(300),
        rating        NUMERIC(5, 2) DEFAULT
0.00,
        category      VARCHAR(30),
        level         VARCHAR(30),
        cost          NUMERIC(8, 2) NOT NULL
DEFAULT 0.00,
        discount_allowed BOOLEAN,
        quiz_threshold  NUMERIC(5, 2),
        PRIMARY KEY (CID),
        FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE
NO ACTION ON UPDATE CASCADE,
        UNIQUE (IID, course_name)
);

CREATE TABLE Section(
        CID           INT NOT NULL,
        section       INT NOT NULL,
        title         VARCHAR(300),
        PRIMARY KEY (CID, section),
        FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE
CASCADE ON UPDATE CASCADE,
        INDEX (section)
);

/* Assumptions:
lecture_content is big data
lecture_content is blob (not recommended, try to store videos in file
directories and hold URLs)
Attention: UNIQUE (CID, title)
*/
CREATE TABLE Lecture(
        CID           INT NOT NULL,
        section       INT NOT NULL,
        content_num   INT NOT NULL,

```

```

        title          VARCHAR(300) NOT NULL,
        lecture_content  VARCHAR(300),
        description      VARCHAR(300),
        PRIMARY KEY (CID, section, content_num),
        FOREIGN KEY (CID, section) REFERENCES Section(CID,
section) ON DELETE CASCADE ON UPDATE CASCADE,
        UNIQUE (CID, title),
        INDEX (section),
        INDEX (content_num)
);

CREATE TABLE Quiz (
    CID           INT NOT NULL,
    section       INT NOT NULL,
    content_num   INT NOT NULL,
    IID           INT NOT NULL,
    title         VARCHAR(300) NOT NULL,
    PRIMARY KEY (CID, section, content_num),
    FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (section) REFERENCES Section(section) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE
NO ACTION ON UPDATE CASCADE,
    UNIQUE (CID, title),
    INDEX (CID),
    INDEX (section),
    INDEX (content_num)
);

/*
Attention: answer ENUM('choice1', 'choice2', 'choice3') NOT NULL DEFAULT
'choice1'

answer = 1 => answer = choice1
*/
CREATE TABLE Quiz_Question(
    CID           INT NOT NULL,
    section       INT NOT NULL,

```

```

        content_num          INT NOT NULL,
        question_num         INT NOT NULL,
        question_text        VARCHAR(300) NOT NULL,
        choice1              VARCHAR(50) NOT NULL
    DEFAULT '' ,
        choice2              VARCHAR(50) ,
        choice3              VARCHAR(50) ,
        answer               ENUM('choice1', 'choice2',
'choice3') NOT NULL DEFAULT 'choice1',
    PRIMARY KEY (CID, section, content_num,
question_num),
    FOREIGN KEY (CID, section, content_num)
REFERENCES Quiz(CID, section, content_num) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE Note(
    Note_ID           INT AUTO_INCREMENT,
    SID               INT NOT NULL,
    section           INT NOT NULL,
    content_num       INT NOT NULL,
    text              VARCHAR(300) NOT NULL,
    PRIMARY KEY (Note_ID),
    FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Course_Announcement(
    Ann_ID            INT AUTO_INCREMENT,
    IID               INT NOT NULL,
    CID               INT NOT NULL,
    date              DATE,
    subject_name      VARCHAR(30) NOT NULL,
    text              VARCHAR(300) NOT
NULL,
    PRIMARY KEY (Ann_ID),
    FOREIGN KEY (IID) REFERENCES
Instructor(IID) ON DELETE CASCADE ON UPDATE CASCADE,

```

```

        FOREIGN KEY (CID) REFERENCES Course(CID)
ON DELETE CASCADE ON UPDATE CASCADE
) ;

CREATE TABLE QnA_Entry_Student(
    QnA_ID_S          INT AUTO_INCREMENT,
    SID                INT NOT NULL,
    CID                INT NOT NULL,
    subject_name       VARCHAR(30) NOT NULL,
    text               VARCHAR(300) NOT NULL,
    date               DATE,
    PRIMARY KEY (QnA_ID_S),
    FOREIGN KEY (SID) REFERENCES Student(SID)
ON DELETE NO ACTION ON UPDATE CASCADE,
        FOREIGN KEY (CID) REFERENCES Course(CID) ON
DELETE CASCADE ON UPDATE CASCADE
) ;

CREATE TABLE QnA_Entry_Instructor(
    QnA_ID           INT AUTO_INCREMENT,
    QnA_ID_S         INT NOT NULL,
    IID              INT NOT NULL,
    CID              INT NOT NULL,
    subject_name     VARCHAR(30) NOT
NULL,
    text             VARCHAR(300) NOT
NULL,
    date             DATE,
    PRIMARY KEY (QnA_ID),
    FOREIGN KEY (QnA_ID_S) REFERENCES
QnA_Entry_Student(QnA_ID_S) ON DELETE NO ACTION ON UPDATE CASCADE,
        FOREIGN KEY (IID) REFERENCES
Instructor(IID) ON DELETE NO ACTION ON UPDATE CASCADE,
        FOREIGN KEY (CID) REFERENCES Course(CID)
ON DELETE CASCADE ON UPDATE CASCADE
) ;

CREATE TABLE Complaint_Entry_Student(

```

```

        Complaint_ID      INT
AUTO_INCREMENT,
        SID      INT NOT NULL,
        CID      INT NOT NULL,
        subject_title  VARCHAR(30) NOT
NULL,
        text     VARCHAR(300) NOT
NULL,
        date     DATE,
        refund_request BOOLEAN,
PRIMARY KEY (Complaint_ID),
FOREIGN KEY (SID) REFERENCES
Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (CID) REFERENCES
Course(CID) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Complaint_Entry_Instructor(
        Complaint_ID      INT
AUTO_INCREMENT,
        IID      INT NOT NULL,
        CID      INT NOT NULL,
        subject_name  VARCHAR(30)
NOT NULL,
        text     VARCHAR(300)
NOT NULL,
        date     DATE,
PRIMARY KEY (Complaint_ID),
FOREIGN KEY (IID) REFERENCES
Instructor(IID) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (CID) REFERENCES
Course(CID) ON DELETE CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Student and Course
Attributes: date

```

```

/*
CREATE TABLE Wishlist(
    SID           INT NOT NULL,
    CID           INT NOT NULL,
    PRIMARY KEY (SID, CID),
    FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE
CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Student and Course
Attributes: progress, rating
*/
CREATE TABLE Enrolls(
    SID           INT NOT NULL,
    CID           INT NOT NULL,
    progress      NUMERIC(5, 2) DEFAULT 0.00,
    date          DATE,
    PRIMARY KEY (SID, CID),
    FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE
CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Student and Course
Attributes: date
*/
CREATE TABLE Certificate(
    SID           INT NOT NULL,
    CID           INT NOT NULL,
    certificate_text  VARCHAR(300),
    rating        NUMERIC(5, 2),
    comment       VARCHAR(300),
    date          DATE,

```

```

        PRIMARY KEY (SID, CID),
        FOREIGN KEY (SID) REFERENCES Student(SID) ON
DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (CID) REFERENCES Course(CID) ON
DELETE CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Instructor and Course
*/
CREATE TABLE Teaches(
    IID                      INT NOT NULL,
    CID                      INT NOT NULL,
    PRIMARY KEY (IID, CID),
    FOREIGN KEY (IID) REFERENCES Instructor(IID) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE
CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Admin and Complaint_Entry
Attention: FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON
UPDATE CASCADE
*/
CREATE TABLE Evaluates_Complaint_Entry_Student(
    AID                      INT NOT
NULL,
    Complaint_ID             INT NOT
NULL,
    response
VARCHAR(300),
    isRefund
BOOLEAN,
    PRIMARY KEY (AID,
Complaint_ID),
    FOREIGN KEY (AID)
REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE,

```

```

        FOREIGN KEY (Complaint_ID)
REFERENCES Complaint_Entry_Student(Complaint_ID) ON DELETE CASCADE ON UPDATE
CASCADE
);

/*
Binary relationship between Admin and Complaint_Entry
Attention: FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON
UPDATE CASCADE
*/
CREATE TABLE Evaluates_Complaint_Entry_Instructor(
    AID           INT
NOT NULL,
    Complaint_ID   INT
NOT NULL,
    response
VARCHAR(300),
    PRIMARY KEY (AID,
Complaint_ID),
    FOREIGN KEY (AID)
REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE,
    FOREIGN KEY
(Complaint_ID) REFERENCES Complaint_Entry_Instructor(Complaint_ID) ON DELETE
CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Admin and Course
Attributes: rate (e.g. %30)
Attention: FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON
UPDATE CASCADE
*/
CREATE TABLE Discount(
    AID           INT NOT NULL,
    CID           INT NOT NULL,
    rate          NUMERIC(5, 2),
    PRIMARY KEY (AID, CID),
    FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO

```

```

ACTION ON UPDATE CASCADE,
    FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE
CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Instructor and Course
Attributes: invite_link, start_date
Attention: it uses its own key to allow instructors to create several links
*/
CREATE TABLE Zoom_Session(
    Session_ID      INT AUTO_INCREMENT,
    IID             INT NOT NULL,
    CID             INT NOT NULL,
    invite_link     VARCHAR(1024),
    start_date      TIMESTAMP,
    PRIMARY KEY (Session_ID),
    FOREIGN KEY (IID) REFERENCES Instructor(IID) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CID) REFERENCES Course(CID) ON
DELETE CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Student and Quiz
Attributes: grade (updated according to Student_Take_Quiz_Question)
Attention: its key is Student(SID) U Quiz(CID, QID)
*/
CREATE TABLE Take_Quiz(
    SID              INT NOT NULL,
    CID              INT NOT NULL,
    section          INT NOT NULL,
    content_num      INT NOT NULL,
    grade            NUMERIC(5, 2),
    PRIMARY KEY (SID, CID, section, content_num),
    FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE
CASCADE ON UPDATE CASCADE
);

```

```

/*
Binary relationship between Student and Quiz_Question
keeps students' answers to each quiz question
Attributes: isTrue
*/
CREATE TABLE Take_Quiz_Question(
    SID           INT NOT NULL,
    CID           INT NOT NULL,
    section       INT NOT NULL,
    content_num   INT NOT NULL,
    question_num  INT NOT NULL,
    answer        ENUM('choice1',
'choice2', 'choice3') NOT NULL,
    isTrue        BOOLEAN,
    PRIMARY KEY (SID, CID, section,
content_num, question_num),
    FOREIGN KEY (SID) REFERENCES Student(SID)
ON DELETE CASCADE ON UPDATE CASCADE
);

/*
Binary relationship between Student and Lecture
keeps students' completion to each lecture (lecture_completed_threshold in
lecture (e.g. %70 for each lecture))
Attributes: isCompleted
*/
CREATE TABLE Take_Lecture(
    SID           INT NOT NULL,
    CID           INT NOT NULL,
    section       INT NOT NULL,
    content_num   INT NOT NULL,
    isCompleted   BOOLEAN,
    PRIMARY KEY (SID, CID, section, content_num),
    FOREIGN KEY (SID) REFERENCES Student(SID) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CID, section, content_num)
REFERENCES Lecture(CID, section, content_num) ON DELETE CASCADE ON UPDATE

```

```
CASCADE
) ;

/*
Example Insertions
 */

INSERT INTO Admin (name, e_mail, password, salary)
VALUES ('omer', 'omer@omer', '1234', 1000000000.00);
```