



# CS 353 COURSE PROJECT

## Design Report

Ucollege - Group 7

Muhammed Emre Yıldız	21702825	Section 3
Murat Angın	21702962	Section 3
Osman Batur İnce	21802609	Section 3
Ümit Yiğit Başaran	21704103	Section 3

**Instructor:** Özgür Ulusoy

**Teaching Assistant:** Mustafa Can Çavdar

April 2, 2020

<b>Revised E/R Model</b>	<b>4</b>
Modifications	4
New Features	5
<b>Relation Schemas</b>	<b>6</b>
Person Schema	6
User Schema	7
Creator Schema	8
Admin Schema	8
Course Schema	9
Buy Schema	10
Lecture Schema	11
Question Schema	12
Answer Schema	13
Refund Schema	14
Discount Schema	15
Note Schema	16
Bookmark Schema	17
CompleteLecture Schema	17
Allow Schema	18
Certificate Schema	18
HasCertificates Schema	19
WishlistCourses Schema	19
CartCourses Schema	20
Achievement Schema	20
Achieve Schema	21
Rating Schema	22
Tag Schema	23
TagsOf Schema	23
Announcement Schema	24
Quiz Schema	25
Participate Schema	26
FlashCard Schema	27
TrueFalse Schema	27
MultipleChoice Schema	28
<b>UI Design and Corresponding SQL Statements</b>	<b>29</b>
Screens From User Perspective	29
Signup Screen	29
Login Screen	31
Filter and List All Courses	33
My Courses	36

User Profile	37
Course Description Page	38
Creator Profile	41
My Cart	42
My Wishlist	43
Add Balance	44
Buy, user have enough balance	46
Buy, user don't have enough balance	48
Course Homepage	48
Lecture Screen	49
Course Quiz Page	52
Quiz Screen	53
Course QA Page	53
Course Ratings Page	55
Course Announcements Page	57
Course About Page	58
Send Refund Request Page	59
Refund Status Page	60
Screens From Creator Perspective	61
Create Course Page	61
Create Lecture Page	62
Create Quiz Page	63
Create Announcement Page	65
Answer Question Page	66
Allow Discount Page	67
Screens From Admin Perspective	68
Create Discount Page	68
Check Refunds Page	69
<b>Implementation Plan</b>	<b>70</b>

# Revised E/R Model

Considering the feedback we got from our TA, we added some new entities, relations, deleted some entities and relations, and updated existing ones to minimize our errors and increase the functionality of the overall system.

## Modifications

- Converted Wishlist, Cart, and Rating entities from weak entities to strong entities
- Updated Wishlist's cardinality in Wishes relation from one to many as a user can have only one wishlist..
- Updated Cart's cardinality in WillBuy relation from one to many.
- Updated Rating's cardinality in RatedBy relation from one to many.
- Updated User's cardinality in RatedBy relation from many to one.
- Changed Certificate\_of relation's name to CertificateOf to preserve the convention.
- Updated Certificate's cardinality in CertificateOf relation from many to one.
- Updated Course's cardinality in CertificateOf relation from many to one
- Updated Course's participation in CertificateOf relation from partial to total
- Converted Announcement entity from weak entity to strong entity.
- Added Makes relationship between Creator and Announcement entities
- Changed the name of User's attribute "isCoursesVisible" to "hideCourses", which hides courses of a user from another user, if it is checked.
- Converted Bookmark, Note, and Lecture entities from weak entities to strong entities.
- Changed the name of Complain entity to Refund.
- Deleted the refundNeeded attribute from Refund entity.

- Renamed Approve relation to Check, and added a state attribute, which are PENDING, APPROVED, REJECTED. At the beginning, a refund request will have the state attribute valued PENDING.
- When Admin "checks" the request and approves it, the state attribute will be APPROVED. If the Admin rejects the request, the state attribute will be REJECTED.
- Converted Answer, Question, Refund and Discount entities from weak entities to strong entities, as they have relations to other strong entities.
- Updated Refund's participation in Request relation from partial to total.
- Updated Discount's participation in Offer relation from partial to total.
- Removed url attribute from Certificate entity as it is only related specified Course entity.
- Updated Rating's participation in Rated relation from partial to total.
- Removed RatedBy relation as it was redundant.
- Added video property to the Lecture entity to hold a video link for the lecture.
- Added seen property to the Refund entity to notify users when a refund request is evaluated.
- Added balance property to User entity to simulate a transaction system with courses.

## **New Features**

- Added Achievement entity contains id, title, and photo attributes.
- Added relationship "Achieve" between User and Achievement entities.
- Added Quiz, Flashcard, TrueFalse, and MultipleChoice entities.
- Added disjoint specialization Flashcard to MultipleChoice and TrueFalse.
- Added HasCard, AddCard, CreateQuiz, and Participate relations.

At the next page there is our revised E/R diagram.



# Relation Schemas

## Person Schema

**Relational Model:** Person(ID, email, password, name, surname, username, photo)

**Candidate Keys:** { {email}, (username), (ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {}

**Table Definition:**

```
CREATE TABLE Person(  
    ID INT AUTO_INCREMENT,  
    email VARCHAR(64) NOT NULL,  
    password VARCHAR(30) NOT NULL,  
    name VARCHAR(20) NOT NULL,  
    surname VARCHAR(20) NOT NULL,  
    username VARCHAR(20) NOT NULL,  
    photo VARCHAR(255),  
    PRIMARY KEY (ID),  
    UNIQUE(email)  
    UNIQUE(username)  
);
```

## User Schema

**Relational Model:** User(ID, hideCourses, balance)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(ID)}

**Table Definition:**

```
CREATE TABLE User( ID INT AUTO_INCREMENT,  
                    hideCourses TINYINT(1) NOT NULL DEFAULT 0,  
                    balance DECIMAL(10,2) NOT NULL DEFAULT 0,  
                    PRIMARY KEY (ID),  
                    FOREIGN KEY ID REFERENCES Person  
);
```



## Creator Schema

**Relational Model:** Creator(ID, about, website, linkedin, youtube)

**Primary Key:** {(ID)}

**Foreign Keys:** {(ID)}

**Table Definition:**

```
CREATE TABLE Creator( ID INT AUTO_INCREMENT,  
                        about VARCHAR(1024),  
                        website VARCHAR(255),  
                        linkedin VARCHAR(255),  
                        youtube VARCHAR(255),  
                        PRIMARY KEY (ID),  
                        FOREIGN KEY ID REFERENCES Person  
);
```

## Admin Schema

**Relational Model:** Admin(ID)

**Primary Key:** {(ID)}

**Foreign Keys:** {(ID)}

**Table Definition:**

```
CREATE TABLE Admin( ID INT AUTO_INCREMENT,  
                     PRIMARY KEY (ID)  
                     FOREIGN KEY ID REFERENCES Person  
);
```

## Course Schema

**Relational Model:** Course (ID, title, price, description, thumbnail, category, creator\_id, averageRating, ratingCount)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(creator\_id)}

**Table Definition:**

```
CREATE TABLE Course( ID INT AUTO_INCREMENT,  
    title VARCHAR(64) NOT NULL,  
    price NUMERIC(6, 2) NOT NULL,  
    description VARCHAR(255) NOT NULL,  
    thumbnail VARCHAR(255) NOT NULL,  
    category VARCHAR(64) NOT NULL,  
    creator_id INT NOT NULL,  
    averageRating FLOAT NOT NULL DEFAULT 0,  
    ratingCount INT NOT NULL DEFAULT 0,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (creator_id) REFERENCES Creator(ID)  
);
```

## Buy Schema

**Relational Model:** Buy(user\_id, course\_id)

**Candidate Keys:** {( user\_id, course\_id )}

**Primary Key:** {( user\_id, course\_id )}

**Foreign Keys:** {(user\_id), (course\_id)}

**Table Definition:**

```
CREATE TABLE Buy( user_id INT,  
                   course_id INT,  
                   PRIMARY KEY (user_id, course_id),  
                   FOREIGN KEY user_id REFERENCES User(ID),  
                   FOREIGN KEY course_id REFERENCES Buy(ID)  
);
```

## Lecture Schema

**Relational Model:** Lecture(ID, chapterName, title,duration, date, isVisible, additionalMaterial, video, course\_id, lecture\_index)

**Candidate Keys:** {(ID), (course\_id, lecture\_index)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(course\_id)}

**Table Definition:**

```
CREATE TABLE Lecture( ID INT AUTO_INCREMENT,  
    chapterName VARCHAR(64) NOT NULL,  
    title VARCHAR(64) NOT NULL,  
    duration TIME(7) NOT NULL,  
    date DATE NOT NULL DEFAULT(getdate()),  
    isVisible TINYINT(1) NOT NULL DEFAULT 1,  
    additionalMaterial VARCHAR(255),  
    video VARCHAR(512),  
    course_id INT NOT NULL,  
    lecture_index INT NOT NULL,  
    PRIMARY KEY (ID),  
    UNIQUE( course_id, lecture_index),  
    FOREIGN KEY (course_id) REFERENCES Course(ID)  
);
```

## Question Schema

**Relational Model:** Question(ID, content, date, user\_id, course\_id, parent\_id)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(user\_id), (course\_id)}

**Table Definition:**

```
CREATE TABLE Question( ID INT AUTO_INCREMENT,  
    content VARCHAR(1024) NOT NULL,  
    date DATETIME NOT NULL DEFAULT(getdate()),  
    user_id INT NOT NULL,  
    course_id INT NOT NULL,  
    parent_id INT DEFAULT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (user_id) REFERENCES User(ID),  
    FOREIGN KEY (course_id) REFERENCES Course(ID)  
);
```

## Answer Schema

**Relational Model:** Answer(ID, content, date, question\_id, creator\_id)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(question\_id), (creator\_id)}

**Table Definition:**

```
CREATE TABLE Answer( ID INT AUTO_INCREMENT,  
    content VARCHAR(1024) NOT NULL,  
    date DATETIME NOT NULL DEFAULT(getdate()),  
    question_id INT NOT NULL,  
    creator_id INT NOT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (question_id) REFERENCES Question(ID),  
    FOREIGN KEY (creator_id) REFERENCES Creator(ID)  
);
```

## Refund Schema

**Relational Model:** Refund(ID, title, reason, user\_id, course\_id, seen, admin\_id, state)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(user\_id, course\_id), (admin\_id)}

**Table Definition:**

```
CREATE TABLE Refund( ID INT AUTO_INCREMENT,  
    title VARCHAR(64) NOT NULL,  
    state VARCHAR(16) NOT NULL DEFAULT 'PENDING' CHECK (state IN ('PENDING',  
'ALLOWED', 'REJECTED')),  
    seen TINYINT(1) NOT NULL DEFAULT 0,  
    reason VARCHAR(1024) NOT NULL,  
    user_id INT NOT NULL,  
    course_id INT NOT NULL,  
    admin_id INT,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (user_id, course_id) REFERENCES Buy(user_id, course_id),  
    FOREIGN KEY (admin_id) REFERENCES Admin(ID)  
);
```

## Discount Schema

**Relational Model:** Discount(ID, percentage, startDate, endDate, course\_id, admin\_id)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(course\_id), (admin\_id)}

**Table Definition:**

```
CREATE TABLE Discount( ID INT AUTO_INCREMENT,  
  
    percentage NUMERIC (5,2) NOT NULL,  
  
    startDate DATETIME NOT NULL,  
  
    endDate DATETIME NOT NULL,  
  
    course_id INT NOT NULL,  
  
    admin_id INT NOT NULL,  
  
    PRIMARY KEY (ID),  
  
    FOREIGN KEY (course_id) references Course(ID)  
  
    FOREIGN KEY (admin_id) REFERENCES Admin(ID)  
  
);
```



## Note Schema

**Relational Model:** Note(ID, title, content, date, user\_id, lecture\_id)

**Primary Key:** {(ID)}

**Foreign Keys:** {(user\_id), (lecture\_id)}

**Table Definition:**

```
CREATE TABLE Note( ID INT AUTO_INCREMENT,  
                    title VARCHAR(64) NOT NULL,  
                    content VARCHAR(512),  
                    date DATETIME NOT NULL DEFAULT(getdate()),  
                    PRIMARY KEY (ID),  
                    FOREIGN KEY (user_id) REFERENCES User(ID),  
                    FOREIGN KEY (lecture_id) REFERENCES Lecture(ID)  
);
```

## Bookmark Schema

**Relational Model:** Bookmark(ID, timestamp, date, user\_id, lecture\_id)

**Primary Key:** {(ID)}

**Foreign Keys:** {(user\_id), (lecture\_id)}

**Table Definition:**

```
CREATE TABLE Bookmark( ID INT AUTO_INCREMENT,  
    timestamp NOT NULL,  
    date  
    PRIMARY KEY (ID),  
    FOREIGN KEY (user_id) REFERENCES User(ID),  
    FOREIGN KEY (lecture_id) REFERENCES Lecture(ID)  
);
```

## CompleteLecture Schema

**Relational Model:** CompleteLecture (lecture\_id, user\_id, course\_id)

**Candidate Keys:** {(lecture\_id, user\_id, course\_id)}

**Primary Key:** {(lecture\_id, user\_id, course\_id)}

**Foreign Keys:** {(lecture\_id), (user\_id, course\_id)}

**Table Definition:**

```
CREATE TABLE CompleteLecture( lecture_id INT,  
    user_id INT  
    course_id INT  
    PRIMARY KEY (lecture_id, user_id, course_id)  
    FOREIGN KEY lecture_id REFERENCES Lecture  
    FOREIGN KEY (user_id, course_id) REFERENCES Buy);
```

## Allow Schema

**Relational Model:** Allow (creator\_id, discount\_id)

**Candidate Keys:** {(creator\_id, discount\_id)}

**Primary Key:** {(creator\_id, discount\_id)}

**Foreign Keys:** {(creator\_id), (discount\_id)}

**Table Definition:**

```
CREATE TABLE Allow( creator_id INT,  
                    discount_id INT  
                    PRIMARY KEY (creator_id, discount_id)  
                    FOREIGN KEY creator_id REFERENCES Creator(ID)  
                    FOREIGN KEY discount_id REFERENCES Discount(ID));
```

## Certificate Schema

**Relational Model:** Certificate (ID, name, course\_id)

**Candidate Keys:** {(ID), (course\_id)}

**Primary Key:** {ID}

**Foreign Keys:** {(ID), (course\_id)}

**Table Definition:**

```
CREATE TABLE certificate(  
    ID INT AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    course_id INT NOT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY course_id REFERENCES Course(ID),  
    UNIQUE(course_id));
```

## HasCertificates Schema

**Relational Model:** HasCertificates (certificate\_id, user\_id)

**Candidate Keys:** {(certificate\_id, user\_id)}

**Primary Key:** {certificate\_id, user\_id}

**Foreign Keys:** {(certificate\_id), (user\_id)}

**Table Definition:**

```
CREATE TABLE HasCertificates(  
    certificate_id INT,  
    user_id INT,  
    PRIMARY KEY(certificate_id, user_id),  
    FOREIGN KEY certificate_id REFERENCES certificate(ID),  
    FOREIGN KEY user_id REFERENCES user(ID));
```

## WishlistCourses Schema

**Relational Model:** WishlistCourses(user\_id, course\_id)

**Candidate Keys:** {(user\_id, course\_id)}

**Primary Key:** {(user\_id, course\_id)}

**Foreign Keys:** {(user\_id), (course\_id)}

**Table Definition:**

```
CREATE TABLE WishlistCourses(  
    user_id INT NOT NULL,  
    course_id INT user_id NULL,  
    PRIMARY KEY(user_id, course_id),  
    FOREIGN KEY user_id REFERENCES User(ID),  
    FOREIGN KEY course_id REFERENCES Course(ID));
```

## CartCourses Schema

**Relational Model:** CartCourses(user\_id, course\_id)

**Candidate Keys:** {(user\_id, course\_id)}

**Primary Key:** {(user\_id, course\_id)}

**Foreign Keys:** {(user\_id), (course\_id)}

**Table Definition:**

```
CREATE TABLE CartCourses(  
    user_id INT NOT NULL,  
    course_id INT user_id NULL,  
    PRIMARY KEY(user_id, course_id),  
    FOREIGN KEY user_id REFERENCES User(ID),  
    FOREIGN KEY course_id REFERENCES Course(ID),  
);
```

## Achievement Schema

**Relational Model:** Achievement(ID, title, photo)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Table Definition:**

```
CREATE TABLE Achievement(  
    ID INT NOT NULL,  
    title VARCHAR(255) NOT NULL,  
    photo VARCHAR(255) NOT NULL,  
    PRIMARY KEY(id));
```

## Achieve Schema

**Relational Model:** Achieve(achievement\_id, user\_id)

**Candidate Keys:** {(achievement\_id, user\_id)}

**Primary Key:** {(achievement\_id, user\_id)}

**Foreign Keys:** {(achievement\_id, user\_id)}

**Table Definition:**

```
CREATE TABLE Achieve(  
    achievement_id INT NOT NULL,  
    user_id INT NOT NULL,  
    PRIMARY KEY(achievement_id, user_id),  
    FOREIGN KEY achievement_id REFERENCES Achievement(ID),  
    FOREIGN KEY user_id REFERENCES User(ID),  
);
```

## Rating Schema

**Relational Model:** Rating(ID, ratingScore, content, date, user\_id, course\_id)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(user\_id, course\_id)}

**Table Definition:**

```
CREATE TABLE Rating(  
    ID INT NOT NULL AUTO_INCREMENT,  
    ratingScore DECIMAL(2,1) NOT NULL,  
    content VARCHAR(512),  
    date DATE NOT NULL DEFAULT(getdate()),  
    user_id INT NOT NULL,  
    course_id INT NOT NULL,  
    PRIMARY KEY(ID),  
    FOREIGN KEY (user_id, course_id) REFERENCES Buy,  
);
```

## Tag Schema

**Relational Model:** Tag(name, color)

**Candidate Keys:** {(name)}

**Primary Key:** {(name)}

**Table Definition:**

```
CREATE TABLE Tag(  
    name VARCHAR(100) NOT NULL,  
    color VARCHAR(64) NOT NULL,  
    PRIMARY KEY(name),  
);
```

## TagsOf Schema

**Relational Model:** TagsOf(tag\_name, course\_id)

**Candidate Keys:** {(tag\_name, course\_id)}

**Primary Key:** {(tag\_name, course\_id)}

**Foreign Keys:** {(course\_id), (tag\_name)}

**Table Definition:**

```
CREATE TABLE TagsOf(  
    tag_name VARCHAR(100) NOT NULL,  
    course_id VARCHAR(100) NOT NULL,  
    PRIMARY KEY(tag_name, course_id),  
    FOREIGN KEY tag_name REFERENCES Tag(name),  
    FOREIGN KEY course_id REFERENCES Course(ID),  
);
```



## Announcement Schema

**Relational Model:** Announcement(ID, title, content, date, creator\_id, course\_id)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(creator\_id), (course\_id)}

**Table Definition:**

```
CREATE TABLE Announcement(  
    ID INT NOT NULL AUTO_INCREMENT,  
    title VARCHAR(100),  
    content VARCHAR(512),  
    date DATE NOT NULL DEFAULT(getdate()),  
    creator_id INT NOT NULL,  
    course_id INT NOT NULL,  
    PRIMARY KEY(ID),  
    FOREIGN KEY creator_id REFERENCES Creator(ID),  
    FOREIGN KEY course_id REFERENCES Course(ID),  
);
```

## Quiz Schema

**Relational Model:** Quiz(ID, duration, name, creator\_id, course\_id)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(creator\_id), (course\_id)}

**Table Definition:**

```
CREATE TABLE Quiz(  
    ID INT NOT NULL,  
    duration TIME(7) NOT NULL,  
    name VARCHAR(128) NOT NULL,,  
    creator_id INT NOT NULL,  
    course_id INT NOT NULL,  
    PRIMARY KEY(ID),  
    FOREIGN KEY creator_id REFERENCES Creator(ID),  
    FOREIGN KEY course_id REFERENCES Course(ID),  
);
```

## Participate Schema

**Relational Model:** Participate(user\_id, quiz\_id, trueCount, date)

**Candidate Keys:** {(user\_id, quiz\_id)}

**Primary Key:** {(user\_id, quiz\_id)}

**Foreign Keys:** {(user\_id), (quiz\_id)}

**Table Definition:**

```
CREATE TABLE Participate(  
    user_id INT NOT NULL,  
    quiz_id INT NOT NULL,  
    trueCount INT NOT NULL,  
    date DATETIME NOT NULL DEFAULT(getdate()),  
    PRIMARY KEY(user_id, quiz_id),  
    FOREIGN KEY user_id REFERENCES User(ID),  
    FOREIGN KEY quiz_id REFERENCES Quiz(ID),  
);
```

## FlashCard Schema

**Relational Model:** FlashCard(ID, question, quiz\_id)

**Candidate Keys:** {(ID), (question, quiz\_id)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(quiz\_id)}

**Table Definition:**

```
CREATE TABLE FlashCard(  
    ID INT NOT NULL,  
    question VARCHAR(1024),  
    quiz_id INT NOT NULL,  
    PRIMARY KEY(ID),  
    FOREIGN KEY quiz_id REFERENCES Quiz(ID),  
    UNIQUE(question, quiz_id)  
);
```

## TrueFalse Schema

**Relational Model:** TrueFalse(ID, answer)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(ID)}

**Table Definition:**

```
CREATE TABLE TrueFalse(  
    ID INT,  
    answer TINYINT(1) NOT NULL,  
    PRIMARY KEY(ID));
```

## MultipleChoice Schema

**Relational Model:** MultipleChoice(ID, choice1, choice2, choice3, choice4, answer)

**Candidate Keys:** {(ID)}

**Primary Key:** {(ID)}

**Foreign Keys:** {(ID)}

**Table Definition:**

```
CREATE TABLE MultipleChoice(  
    ID INT NOT NULL,  
    choice1 VARCHAR(255) NOT NULL,  
    choice2 VARCHAR(255) NOT NULL,  
    choice3 VARCHAR(255) NOT NULL,  
    choice4 VARCHAR(255) NOT NULL,  
    answer VARCHAR(255) NOT NULL CHECK (answer IN (choice1, choice2, choice3,  
choice4)) ,  
    PRIMARY KEY(ID),  
);
```

# UI Design and Corresponding SQL Statements

## Screens From User Perspective

### Signup Screen

Ucollege Courses My Courses About Log in Sign Up

### Sign up


Username

Email

First Name Last Name

Password

User ☒ Creator

 Change Profile Photo

Sign In

- Sign Up

**INSERT INTO** Person (username, email, name, surname, password, photo)

**VALUES** (<written\_username>, <written\_email>, <written\_name>, <written\_surname>,  
<written\_password>, <selected\_photo>);

- If selected creator, add to creator table

```
INSERT INTO Creator (ID)
```

```
SELECT ID
```

```
FROM Person
```

```
WHERE email = <written_email> AND password = <written_password>;
```

- If selected user, add to user table

```
INSERT INTO User (ID)
```

```
SELECT ID
```

```
FROM Person
```

```
WHERE email = <written_email> AND password = <written_password>;
```

## Login Screen

Ucollege

Courses My Courses About

Log in Sign Up

### Login

New user? [Create an account](#)

Username

Password

☒ Keep me signed in

Sign In

- Check if a person is a user or not

SELECT \*

FROM User U, (SELECT ID FROM Person WHERE username = <written\_username> AND password = <written\_password>) as P

WHERE U.ID = P.ID;



- Check if a person is a creator or not

SELECT \*

FROM Creator C, (SELECT ID FROM Person WHERE username = <written\_username> AND password = <written\_password>) as P

WHERE C.ID = P.ID;

- Check if a person is a admin or not

SELECT \*

FROM Admin A, (SELECT ID FROM Person WHERE username = <written\_username> AND password = <written\_password>) as P

WHERE A.ID = P.ID;

## Filter and List All Courses

Ucollege
Courses
My Courses
About
Log in
Sign Up

# Courses

You can filter and select best courses which available and beneficial for you, lets start.

### Categories

☒ Technology
☐ Cooking
☒ Music
☐ Economy
☒ Social
☒ Life

### Filtering

By price

45TL
120TL

By keyword

Value

### Ordering

Order by

Price
Discount
Rating

Technology | 4.5 / 5 | -%5 Discount | 129.99TL

### AWS Technology Stack

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.

Music | 4.8 / 5 | -%5 Discount | 129.99TL

### Piano Masterclass

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.

Coding | 4.1 / 5 | -%5 Discount | 129.99TL

### Learn Web development

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.

Social | 3.1 / 5 | -%5 Discount | 129.99TL

### Communication Skills

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.

Planning | 2.1 / 5 | -%5 Discount | 129.99TL

### Goal Setting to Success

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.

Life | 2.8 / 5 | -%5 Discount | 129.99TL

### Life Coaching Tips

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.

- Select all courses to list

SELECT \*

FROM Course c, Discount d

WHERE d.course\_id = c.ID;

- Select courses by their categories

SELECT \*

FROM Course

WHERE category IN (<selected\_category\_list>);

- Select courses by their price ranges

```
SELECT *  
  
FROM Course  
  
WHERE price BETWEEN <lower_price> AND <upper_price>;
```

- Select courses by keywords

```
SELECT *  
  
FROM Course  
  
WHERE description LIKE "%<written_keyword>%" OR title LIKE "%<written_keyword>%" OR  
category LIKE "%<written_keyword>%";
```

- Order courses by their prices

```
SELECT *  
  
FROM Course  
  
ORDER BY price DESC;
```

```
SELECT *  
  
FROM Course  
  
ORDER BY price;
```

- Order courses by their discount percentages

```
SELECT c.ID, c.title, c.description,  
c.price, c.thumbnail, c.category, c.averageRating, c.ratingCount  
FROM Course c, inDiscount i, Discount d  
WHERE i.ID = d.ID, i.id = c.id  
ORDER BY d.percentage DESC;
```

```
SELECT c.ID, c.title, c.description,  
c.price, c.thumbnail, c.category, c.averageRating, c.ratingCount  
FROM Course c, inDiscount i, Discount d  
WHERE i.ID = d.ID, i.id = c.id  
ORDER BY d.percentage;
```

- Order courses by their ratings

```
SELECT *  
FROM Course  
ORDER BY averageRating DESC;
```

```
SELECT *  
FROM Course  
ORDER BY averageRating;
```


## My Courses

Ucollege

CoursesMy CoursesAbout

My Profile


# My courses



Technology | October 2019

### AWS Technology Stack


Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.



Music | January 2019

### Piano Masterclass

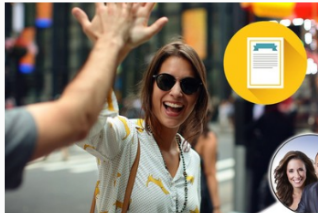
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.



Coding | February | 2020


### Learn Web development

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia in deserunt mollit anim id est sint laborum.




Social | October 2019

### Communication Skills



Planning | October 2019

### Goal Setting to Success



Life | March 2019

### Life Coaching Tips

- List all the courses purchased by a user

**SELECT** id, title, description, thumbnail, category, date

**FROM** Course, Buy

**WHERE** course\_id = id **AND** user\_id = @user\_id;

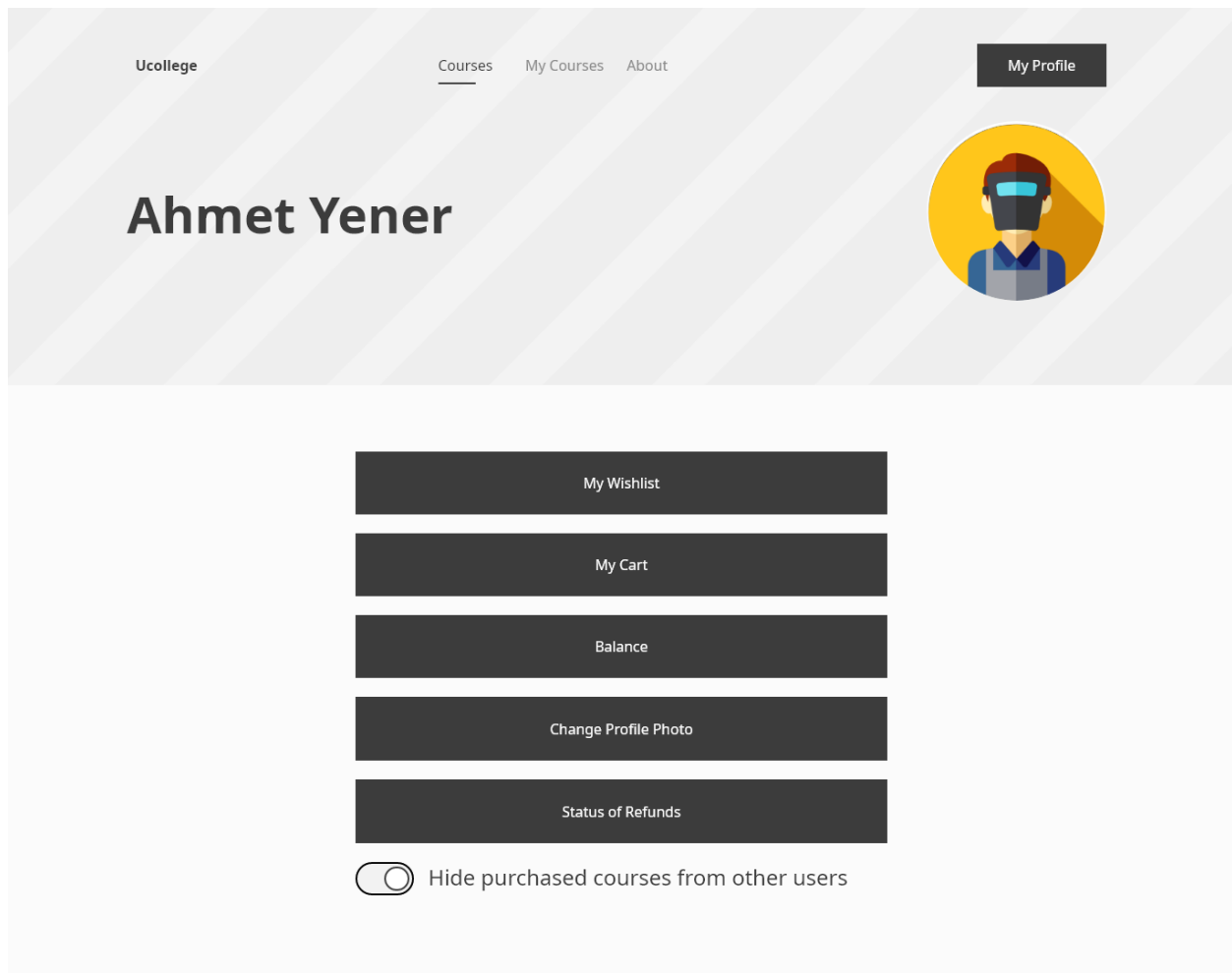
- Get the user photo

**SELECT** photo

**FROM** Person

**WHERE** id = @user\_id;

## User Profile



- Retrieve current person/user

**SELECT** p.name, p.surname, p.photo, p.username

**FROM** User u, Person p

**WHERE** u.ID = p.ID **AND** ID = @user\_id;

- Update user's hide preference

**UPDATE** User

**SET** hideCourses = = 1 - hideCourses

**WHERE** ID = @user\_id;

## Course Description Page


Ucollege
Courses
My Courses
About

My Profile

Technology

# AWS Technology Stack

Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams



### About course

This Ultimate Exam Training for the AWS Certified Cloud Practitioner is packed with comprehensive video lessons, hands-on labs, practice exams, quizzes and exam-crams! If you are new Amazon Web Services / Cloud Computing and looking to confidently pass your AWS Cloud Practitioner Certification Exam first time - then this value-packed AWS training course is for you!

**WHY THIS ULTIMATE EXAM PREP IS YOUR BEST CHANCE TO MAXIMISE YOUR EXAM SUCCESS**

**HIGHLY FLEXIBLE COURSE STRUCTURE:** We understand that not everyone has the time to go through lengthy lectures. That's why we give you options to maximize your time efficiency and accommodate different learning styles

### Ratings

★ 4.3 / 5 Average

Mehmet A.  
It is a really impressive course, you need to buy it.

★ 4.8 / 5

Tom J.  
I like the tone of creator but there is not much information.

★ 2.8 / 5

Normal Price:	120.99TL
Discount Amount:	12.99TL
Current Price:	108.00TL

Buy Course

Add to cart
Add to wishlist



**Jonathan Warner**  
Software Engineer

- Retrieve the course info

```
SELECT title, price, description, thumbnail, category, averageRating  
FROM Course  
WHERE ID = @course_id;
```

- Retrieve course discount

```
SELECT percentage  
FROM Discount  
WHERE course_id = @course_id;
```

- Retrieve course ratings

```
SELECT ID, ratingScore, content  
FROM Rating  
WHERE course_id = @course_id;
```

- Retrieve creators of course comments' name and ID

```
SELECT name, ID  
FROM Person, (SELECT * FROM Rating WHERE course_id = @course_id;)  
WHERE Person.ID = Rating.user_id;
```



- Retrieve course creator

```
SELECT *  
  
FROM Course course, Creator creator, Person person  
  
WHERE course.ID = @course_id AND creator.ID = course.creator_id AND creator.ID = person.ID;
```

- Add the course to cart

```
INSERT INTO CartCourses  
  
VALUES(@course_id, @user_id);
```

- Add the course to wishlist

```
INSERT INTO WishlistCourses  
  
VALUES(@course_id, @user_id);
```

## Creator Profile

The screenshot shows a web page for a creator named Jonathan Warner. At the top, there is a navigation bar with the logo 'Ucollege' and links for 'Courses', 'My Courses', and 'About'. On the right side of the navigation bar are 'Log in' and 'Sign Up' buttons. The main header area features the name 'Jonathan Warner' in a large, bold font, with 'Software Engineer' written below it. To the right of the name is a circular profile picture of Jonathan Warner, a man with glasses and a bow tie. Below the header, there is a section titled 'About creator' which contains three paragraphs of text about his background as a founder of Digital Cloud Training, his experience in IT, and his passion for AWS certification training. At the bottom of this section are three circular icons representing social media links for GitHub, LinkedIn, and YouTube.

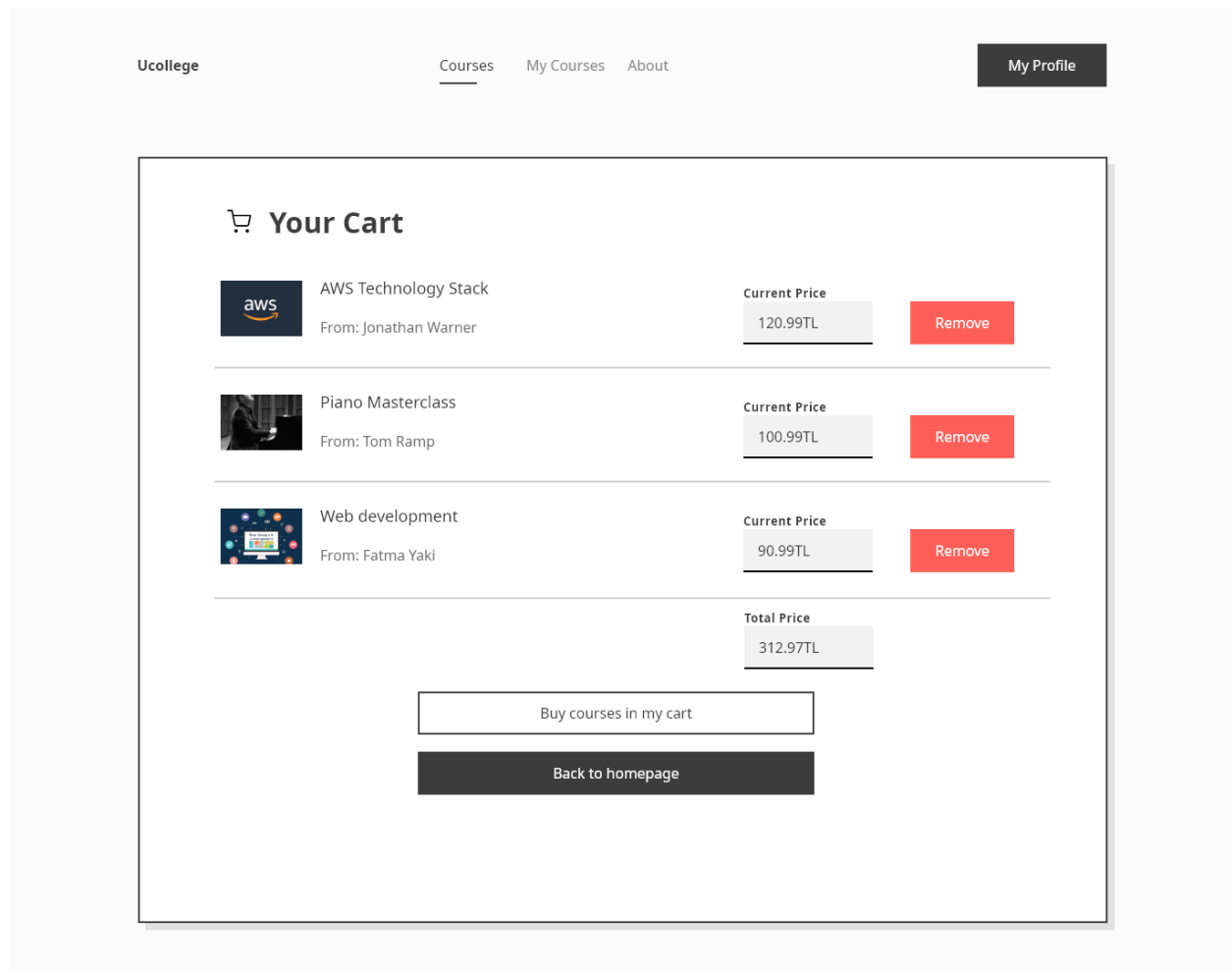
- Show the creator profile

**SELECT** \*

**FROM** Person p, Creator c

**WHERE** p.ID = c.ID **AND** c.ID = <selected\_course\_creator>;

# My Cart



- Retrieve courses in a user's cart

```
SELECT c.title, c.description, c.price, c.thumbnail, c.category, c.averageRating, c.ratingCount
```

```
FROM Course c, CartCourses cart
```

```
WHERE c.ID = cart.course_id AND cart.user_id = @ID;
```

- Buy the courses

```
INSERT INTO Buy(course_id, user_id)
```

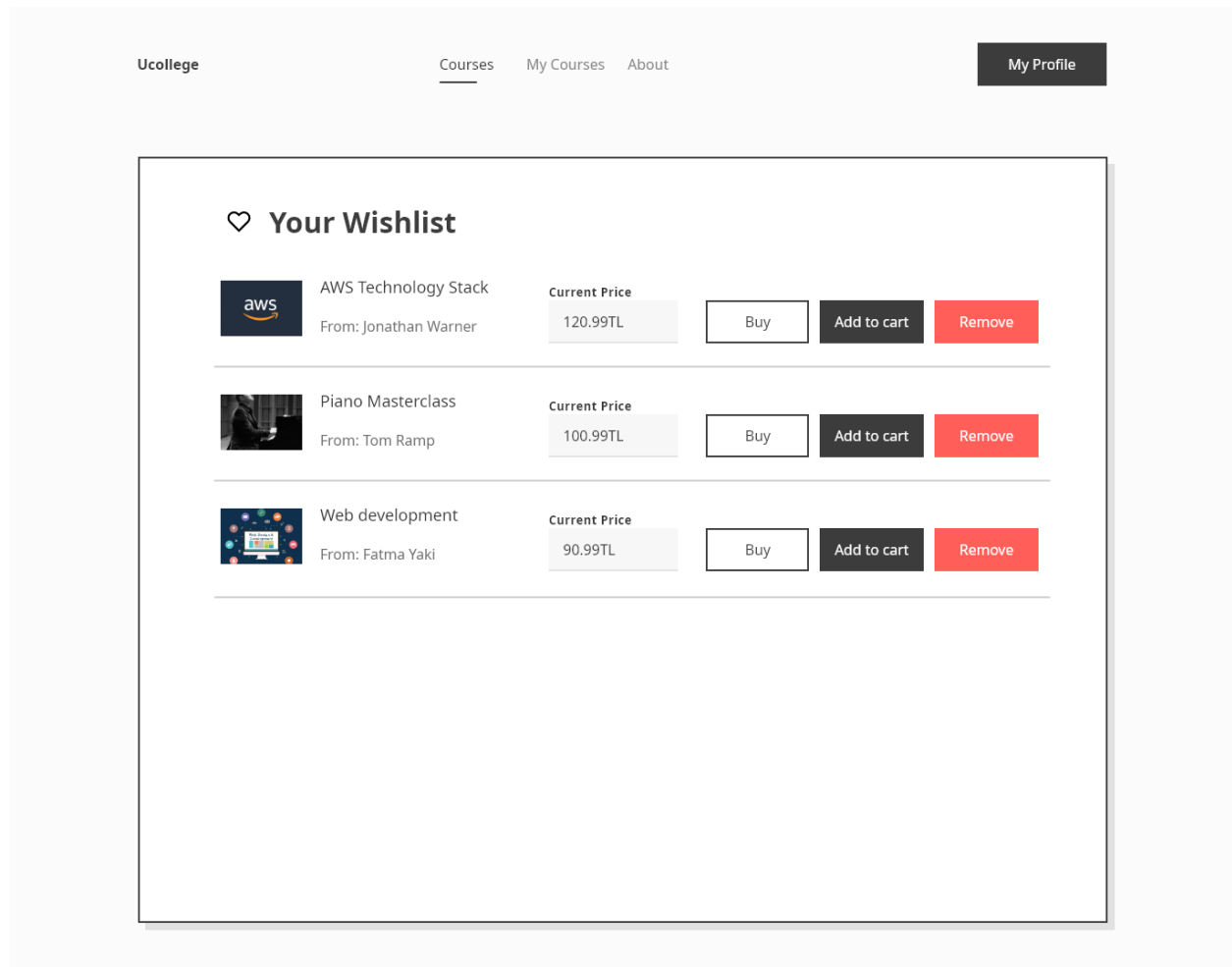
```
VALUES(@course_id, @user_id);
```

- Remove a course from cart

**DELETE FROM** CartCourses

**WHERE** course\_id = @course\_id);

## My Wishlist



- Retrieve courses exists in a user's wishlist

**SELECT** c.title, c.description, c.price, c.thumbnail, c.category, c.averageRating, c.ratingCount

**FROM** Course c, WishlistCourses w

**WHERE** c.ID = w.course\_id **AND** w.user\_id = @ID;

- Buy the courses

```
INSERT INTO Buy(course_id, user_id)
```

```
VALUES(@course_id, @user_id);
```

- Add a course into cart

```
INSERT INTO CartCourses
```

```
VALUES(@course_id, @user_id);
```

## Add Balance

Ucollege

Courses

My Courses

About

My Profile

Ahmet Yener

Current balance: 245TL

25TL

Selected

50TL

Select

100TL

Select

Custom Amount

Select

Card number

CVV

Date

Add Balance

- Retrieve current person

```
SELECT *  
  
FROM User u, Person p  
  
WHERE u.ID = p.ID AND ID = @user_id;
```

- Update the balance

```
UPDATE User  
  
SET balance = balance + <selected_price>  
  
WHERE ID = @user_id;
```


Buy, user have enough balance




Ucollege

[Courses](#)[My Courses](#)[About](#)

My Profile

## Purchase

 You will purchase following courses

	AWS Technology Stack From: Jonathan Warner	Current Price 120.99TL
	Piano Masterclass From: Tom Ramp	Current Price 100.99TL
	Web development From: Fatma Yaki	Current Price 90.99TL

Total Price: 312.99TL

Discount Amount: 12.99TL

Amount to be paid: 300.00TL

Current Balance: 400.00TL

Balance left after buy: 100.00TL

Buy

Back to homepage

- Retrieve courses from cart

**SELECT** c.title, c.description, c.price, c.thumbnail, c.category, c.averageRating, c.ratingCount

**FROM** Course c, CartCourses cart

**WHERE** c.ID = cart.course\_id **AND** cart.user\_id = @ID;

- Find the discounted price

**WITH** tempTable(discountedPrice)

(**SELECT** c.price\*(100-d.percentage)

**FROM** Course c, Discount d, Allow a

WHERE c.id == d.course\_id AND d.id == a.discount\_id AND c.id == @selected\_course\_id AND  
d.course\_id == @selected\_course\_id)

SELECT c.title, c.description, tempTable.discountedPrice, c.thumbnail, c.category, c.averageRating,  
c.ratingCount

FROM Course c, Discount d, Allow a

WHERE c.id == d.course\_id AND d.id == a.discount\_id

- Retrieve current balance

SELECT balance

FROM User

WHERE ID = @user\_id;

- Buy the courses

INSERT INTO Buy(course\_id, user\_id)

VALUES(@course\_id, @user\_id);

- Update the balance

UPDATE User

SET balance = balance - <total\_price>

WHERE ID = @user\_id;



Buy, user don't have enough balance

## Course Homepage

The screenshot shows the 'AWS Technology Stack' course page on the Ucollege platform. The page has a header with 'Ucollege' and navigation links for 'Courses', 'My Courses', and 'About'. A 'My Profile' button is in the top right. The course title 'AWS Technology Stack' is prominently displayed, along with a description: 'Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams'. An AWS logo is featured on the right. Below the title, there's a navigation bar with links for 'Lectures', 'Quizzes', 'Q/A', 'Ratings', 'Announcements', and 'About'. The 'Lectures' section is active, showing a list of five lectures. The first two lectures, 'Introduction' and 'How to install AWS', are marked as 'Completed'. The remaining three lectures, 'Hello world', 'Benefits of AWS', and 'AWS versus Google Cloud', are marked as 'Watch'.

Lectures	Quizzes	Q/A	Ratings	Announcements	About
1. Introduction What is AWS, how to use it					Completed
2. How to install AWS How can you install AWS on your computer easily					Completed
3. Hello world Lets write first example of AWS					Watch
4. Benefits of AWS Lets consider benefits of AWS, cloud systems					Watch
5. AWS versus Google Cloud See the differences between Google Cloud and AWS					Watch

- Retrieve course info

SELECT \*

FROM Course

WHERE ID = @course\_id;

- Retrieve course lectures

SELECT \*

FROM Lecture

WHERE course\_id = @course\_id;

- Retrieve completed lectures

SELECT \*

FROM CompleteLecture cl

WHERE cl.course\_id = @course\_id AND cl.user\_id = @user\_id;

## Lecture Screen

The screenshot shows the Ucollege interface for a lecture titled 'Introduction'. The header includes the Ucollege logo, navigation links for Courses, My Courses (active), and About, and a My Profile button. The main content area displays the chapter title 'Introduction' with a subtitle 'What is AWS, how to use it'. To the right, it shows the chapter name 'Introduction to AWS', duration '9:20', and date added '23 October 2020'. Below this is a video player with a play button. To the right of the video is a note-taking area with a text box labeled 'Your note...' and an 'Add Note' button. At the bottom, there are buttons for 'Previous Lecture', 'Complete Lecture', and 'Next Lecture'. Below the 'Complete Lecture' button is a progress bar showing '02:32' and an 'Add bookmark' button. On the far right, there is a section for 'Additional Material' with a link to a 'Google Drive zip file'.

Ucollege Courses My Courses About My Profile

AWS Technology Stack

# Introduction

What is AWS, how to use it

Chapter Introduction to AWS Duration 9:20 Date added 23 October 2020

Video

Your note...

Add Note

Additional Material

[Google Drive zip file](#)

Previous Lecture Complete Lecture Next Lecture

02:32 Add bookmark

- Retrieve course info

```
SELECT *  
  
FROM Course  
  
WHERE ID = @course_id;
```

- Retrieve the lecture

```
SELECT *  
  
FROM Lecture  
  
WHERE course_id = @course_id AND ID = @lecture_id;
```

- Finish the lecture, then proceed to the next lecture

```
INSERT INTO CompleteLecture (lecture_id, user_id, course_id)  
  
VALUES (@lecture_id, @user_id, @course_id);  
  
SELECT *  
  
FROM Lecture  
  
WHERE lecture_index = (select min(lecture_index) FROM foo WHERE id > @lecture_id);
```

- Proceed to the next lecture, if table is empty there is no next lecture

```
SELECT *  
  
FROM Lecture  
  
WHERE course_id = @course_id AND lecture_index > @lecture_index  
  
ORDER BY lecture_index LIMIT 1;
```

- Proceed to the previous lecture, if table is empty there is no previous lecture

`SELECT *`

`FROM Lecture`

`WHERE course_id = @course_id AND lecture_index < @lecture_index`

`ORDER BY lecture_index DESC LIMIT 1;`

- Create notes on lectures (Visible only to user)

`INSERT INTO Note (title, content, user_id)`

`VALUES (<written_title>, <written_content>, @user_id);`

- Create bookmarks on the lecture

`INSERT INTO Bookmark(timestamp)`

`VALUES (<written_timestamp>);`

## Course Quiz Page

The screenshot shows the 'Ucollege' interface for the 'AWS Technology Stack' course. The top navigation bar includes 'Courses', 'My Courses', 'About', and 'My Profile'. The course title 'AWS Technology Stack' is displayed with a subtitle 'Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams'. The AWS logo is on the right. Below the course title, there are tabs for 'Lectures', 'Quizzes', 'Q/A', 'Ratings', 'Announcements', and 'About'. The 'Quizzes' tab is active, showing a list of four quizzes:

Quiz ID	Quiz Title	Description	Action
1.	Beginner Quiz	What is AWS, how to use it	Score 5/10
2.	Intermediate Quiz	How can you install AWS on your computer easily	Score 9/10
3.	Chapter 1 Quiz	Lets write first example of AWS	Take
4.	Expert Quiz	Lets consider benefits of AWS, cloud systems	Take

- Retrieve quizzes of a course

**SELECT \***

**FROM** Quiz

**WHERE** course\_id = @course\_id;

- Retrieve completed quiz data for that user

**SELECT \***

**FROM** Participate p, (**SELECT** ID **FROM** Quiz **WHERE** course\_id = @course\_id) **AS** qids

**WHERE** p.quiz\_id = qids.ID **AND** p.user\_id = @user\_id;

## Quiz Screen

## Course QA Page

The screenshot shows the 'AWS Technology Stack' course page on the Ucollege platform. The header includes the Ucollege logo, navigation links for 'Courses', 'My Courses', and 'About', and a 'My Profile' button. The course title 'AWS Technology Stack' is prominently displayed, along with a description: 'Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams'. A large AWS logo is featured on the right. Below the header, a navigation bar contains links for 'Lectures', 'Quizzes', 'Q/A' (which is underlined), 'Ratings', 'Announcements', and 'About'. The main content area displays two questions and their answers. The first question, by Ahmet Bilir, asks for the AWS Suite download link, with the answer by Jonathan Warner stating it can be downloaded from a website. The second question, by Mehmet Okur, asks how to uninstall the suite from Windows, with the answer by Jonathan Warner stating it can be done using Ccleaner. On the right side of the main content area, there is a section titled 'Ask a question' with a text input field and an 'Ask' button.

- Retrieve course info

**SELECT** \*

**FROM** Course

**WHERE** ID = @course\_id;

- Retrieve questions and their users

```
SELECT *  
  
FROM Question  
  
WHERE course_id = @course_id;
```

- For each question, we have to display writer's names

```
SELECT ID, name, surname  
  
FROM Person  
  
WHERE ID = @user_id;
```

- Retrieve parent of the questions

```
SELECT Q.id  
  
FROM Question Q  
  
WHERE Q.parent_id = @question_id;
```

- Retrieve answers of the questions

```
SELECT *  
  
FROM Answer  
  
WHERE question_id = @question_id;
```

## Course Ratings Page

The screenshot shows the 'Ucollege' website interface. At the top, there are navigation links: 'Courses', 'My Courses' (underlined), and 'About'. A 'My Profile' button is in the top right. The main header area includes the text 'Technology' and 'AWS Technology Stack' in large bold letters. Below this, a subtitle reads: 'Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams'. To the right is a large image of the AWS logo. A secondary navigation bar contains links: 'Lectures', 'Quizzes', 'Q/A', 'Ratings' (underlined), 'Announcements', and 'About'. The 'Ratings' section displays a '4.3 / 5 Average' with a star icon. Below this, there are six individual ratings from users 'Mehmet A.' and 'Tom J.', each with a comment and a star rating box showing '4.8 / 5' and '2.8 / 5' respectively. To the right of the ratings is a 'Add your rating' form with a star selection bar (0 to 5 stars), a text area for 'Rating comment...', and a 'Rate' button.

- Retrieve ratings of courses

**SELECT** \*

**FROM** Rating

**WHERE** course\_id = @course\_id;

- Retrieve creators of course comments' name and ID

**SELECT** name, ID

**FROM** Person, (**SELECT** \* **FROM** Rating **WHERE** course\_id = @course\_id)

**WHERE** Person.ID = Rating.user\_id;



- Check if the course is finished for the user

```
(SELECT L2.ID
```

```
FROM Lecture L2
```

```
WHERE course_id = @course_id)
```

```
EXCEPT
```

```
(SELECT CL.lecture_id as ID
```

```
FROM CompleteLecture CL
```

```
WHERE course_id = @course_id AND user_id = @user_id)
```

- Add rating if course is finished

```
INSERT INTO Rating(rating_score, content, user_id, course_id)
```

```
VALUES(<rating_score>, <content>, @user_id, @course_id);
```

## Course Announcements Page

The screenshot shows the 'AWS Technology Stack' course page on the 'Ucollege' platform. The page has a header with navigation links: 'Ucollege', 'Courses', 'My Courses' (underlined), and 'About'. A 'My Profile' button is in the top right. Below the header, the course title 'AWS Technology Stack' is displayed in large bold text, with a subtitle 'Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams'. To the right is a large AWS logo. A secondary navigation bar contains links for 'Lectures', 'Quizzes', 'Q/A', 'Ratings', 'Announcements' (underlined), and 'About'. The main content area lists three announcements, each with a title, a description, and a date, separated by horizontal lines:

- Server files updated**  
You need to redownload files in lecture 25  
20 October 2020
- Image asses changed**  
You need to change photos in lecture 21  
13 October 2020
- Password for rar file changed**  
Rar file password at lecture 18 will be awscom.com  
10 October 2020

- Retrieve course data

**SELECT** \*

**FROM** Course

**WHERE** ID = @course\_id;

- Retrieve announcements

**SELECT** title, content, date

**FROM** Announcement

**WHERE** course\_id = @course\_id;


## Course About Page

[Ucollege](#)[Courses](#)[My Courses](#)[About](#)[My Profile](#)

Technology

# AWS Technology Stack

Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams



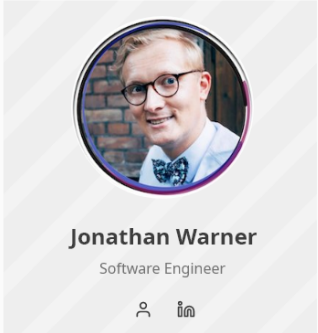
[Lectures](#)[Quizzes](#)[Q/A](#)[Ratings](#)[Announcements](#)[About](#)

### About course

This Ultimate Exam Training for the AWS Certified Cloud Practitioner is packed with comprehensive video lessons, hands-on labs, practice exams, quizzes and exam-crams! If you are new Amazon Web Services / Cloud Computing and looking to confidently pass your AWS Cloud Practitioner Certification Exam first time - then this value-packed AWS training course is for you!

**WHY THIS ULTIMATE EXAM PREP IS YOUR BEST CHANCE TO MAXIMISE YOUR EXAM SUCCESS**

**HIGHLY FLEXIBLE COURSE STRUCTURE:** We understand that not everyone has the time to go through lengthy lectures. That's why we give you options to maximize your time efficiency and accommodate different learning styles



**Jonathan Warner**  
Software Engineer

[Refund](#)

- Retrieve course info

**SELECT \***

**FROM** Course

**WHERE** ID = @course\_id;

- Retrieve creator info

**SELECT \***

**FROM** Creator c, Person p

**WHERE** c.ID = p.ID **AND** c.ID = @creator\_id;

## Send Refund Request Page

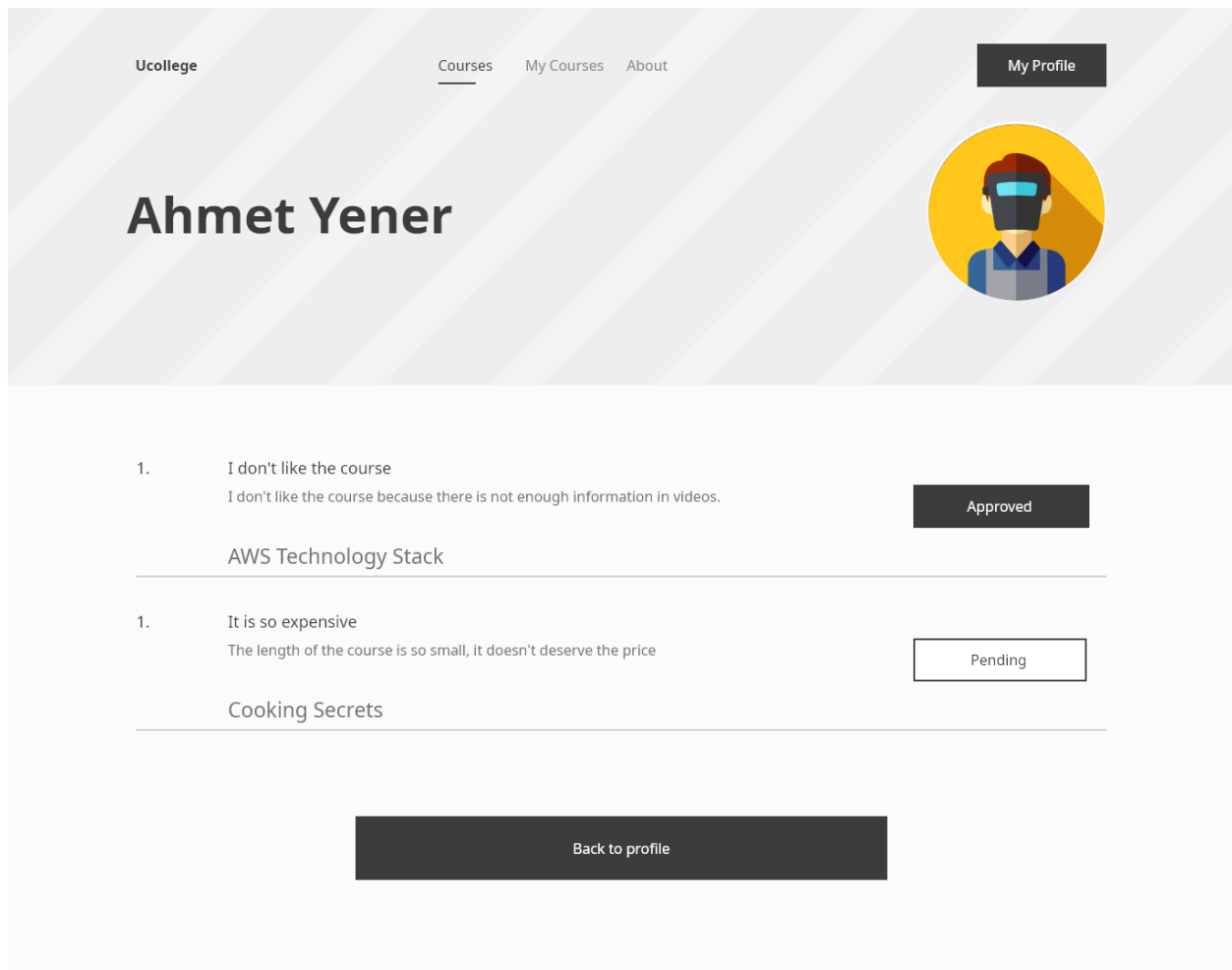
The screenshot shows a web page for 'Ucollege' with a navigation bar containing 'Courses', 'My Courses', and 'About'. A 'My Profile' button is in the top right. The main content area is titled 'Technology' and 'AWS Technology Stack', with a sub-header 'Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams'. An AWS logo is on the right. Below this is a 'Request a refund' form with a 'Title' input field, a 'Reason' text area, and a 'Refund' button.

- Specify the reason for the request and send

```
INSERT INTO Refund(title, reason, user_id, course_id, admin_id)
```

```
VALUES (<title>, <reason>, @user_id, @course_id, NULL);
```

## Refund Status Page



- List all refunds a user requested

**SELECT** \*

**FROM** Refund ref, Course c, Person p

**WHERE** ref.course\_id = c.ID **AND** ref.user\_id = @user\_id **AND** p.id = @user\_id;

# Screens From Creator Perspective

## Create Course Page

The screenshot shows the 'Create Course' form within the Ucollege app. The app's header includes the 'Ucollege' logo, navigation links for 'Courses', 'My Courses', and 'About', and a 'My Profile' button. The form itself is titled 'Create Course' and contains several input fields: 'Title', 'Price', and a large 'Description' text area. Below these fields, there is a 'Change Thumbnail Photo' button next to a placeholder image of a person wearing a VR headset, and a 'Category' dropdown menu. At the bottom of the form is a large 'Create Course' button. A close button (an 'x' in a circle) is located in the top right corner of the form's modal window.

- Create a new course

```
INSERT INTO Course(title, price, description, thumbnail, category, creator_id)
```

```
VALUES (<written_title>, <written_price>, <written_description>, <selected_thumbnail>,  
<selected_category>, @creator_id);
```

## Create Lecture Page

The screenshot shows the 'Create Lecture' form within the Ucollege application. The form is titled 'Create Lecture' and includes a close button (X) in the top right corner. It contains several input fields: 'Chapter name', 'Title', 'Video', 'Video duration', and 'Additional Material'. There is also a toggle switch for 'Is Visible'. A dark 'Create Lecture' button is at the bottom of the form. The background shows the Ucollege navigation bar with links for 'Courses', 'My Courses', and 'About', and a 'My Profile' button.

- Create lectures for the course

**INSERT INTO** Lecture (chapterName, title, duration, isVisible, additionalMaterial, video, course\_id)

**VALUES** (<written\_chapterName>, <written\_title>, <get\_duration >, <selected\_visible>,  
<written\_link\_additionalMaterial>, <written\_link\_video>, @course\_id)

## Create Quiz Page

Ucollege

CoursesMy CoursesAbout

My Profile

Create Quiz

Name

Duration

Questions

1

Question

Answer

True  
False

☐ Multiple  
☐ Choice

2

Question

Answer

True  
False

☐ Multiple  
☐ Choice

Choice 1

Choice 2

Choice 3

Choice 4

3

Question

Answer

True  
False

☐ Multiple  
☐ Choice

Create Quiz

- Insert quiz

```
INSERT INTO Quiz(duration, name, creator_id, course_id)
```

```
VALUES (<written_duration>, <written_name>, @creator_id, @course_id);
```

- Insert questions

```
INSERT INTO FlashCard(question, quiz_id)
```

```
SELECT <written_question>, ID
```

```
FROM Quiz
```

```
WHERE creator_id = @creator_id AND course_id = @course_id;
```



- Insert true false

```
INSERT INTO TrueFalse(ID, answer)

SELECT ID, <selected_answer>

FROM FlashCard f

WHERE f.question = <written_question> AND f.quiz_id IN (SELECT q.ID

FROM Quiz q

WHERE q.creator_id = @creator_id AND q.course_id = @course_id);
```

- Insert multiple choice

```
INSERT INTO MultipleChoice(ID, choice1, choice2, choice3, choice4, answer)

SELECT ID, <written_choice1>, <written_choice2>, <written_choice3>, <written_choice4>,

<selected_answer>

FROM FlashCard f

WHERE f.question = <written_question> AND f.quiz_id IN (SELECT q.ID

FROM Quiz q

WHERE q.creator_id = @creator_id AND q.course_id = @course_id);
```

## Create Announcement Page

The screenshot shows a web application interface for creating an announcement. At the top, there is a navigation bar with the logo 'Ucollege' on the left, and links for 'Courses', 'My Courses', and 'About' in the center. On the right of the navigation bar is a dark button labeled 'My Profile'. Below the navigation bar, a modal window titled 'Create Announcement' is displayed. This modal has a close button (an 'x' in a circle) in the top right corner. Inside the modal, there is a text input field labeled 'Title' and a larger text area labeled 'Content'. At the bottom of the modal is a dark button labeled 'Create Lecture'.

- Insert announcement

```
INSERT INTO Announcement(title, content, creator_id, course_id)
```

```
VALUES (<written_title>, <written_content>, @creator_id, @course_id);
```

## Answer Question Page

Ucollege

Courses

My Courses

About

My Profile

Answer question

Answer

Answer

- Answer the questions

**INSERT INTO** Answer(content, question\_id, creator\_id)

**VALUES** (<written\_content>,@question\_id, @user\_id);

## Allow Discount Page

Ucollege Courses My Courses About My Profile

Technology

# AWS Technology Stack

Ace your AWS Certified Cloud Practitioner exam! Includes Amazon Web Services Certified Cloud Practitioner Practice Exams

Lectures Quizzes Q/A Ratings About Manage Discounts

1.	Discount ID %12 Discount	12 December 2019 - 12 January 2020	Allowed
1.	Discount ID %8 Discount	22 December 2019 - 22 January 2020	Not allowed

- Retrieve discounts

**SELECT** \*

**FROM** Discount d, Course c

**WHERE** d.course\_id = c.ID **AND** c.creator\_id = @creator\_id;

- Insert allow

**INSERT INTO** Allow(creator\_id, discount\_id)

**VALUES**(@creator\_id, @discount\_id);

- Delete from allow

**DELETE FROM** Allow

**WHERE** creator\_id = @creator\_id **AND** discount\_id = @discount\_id;

## Screens From Admin Perspective

### Create Discount Page

The screenshot shows the 'Create Discount' form within the Ucollege admin interface. The form is titled 'Create Discount' and has a close button (X) in the top right corner. It contains the following elements:

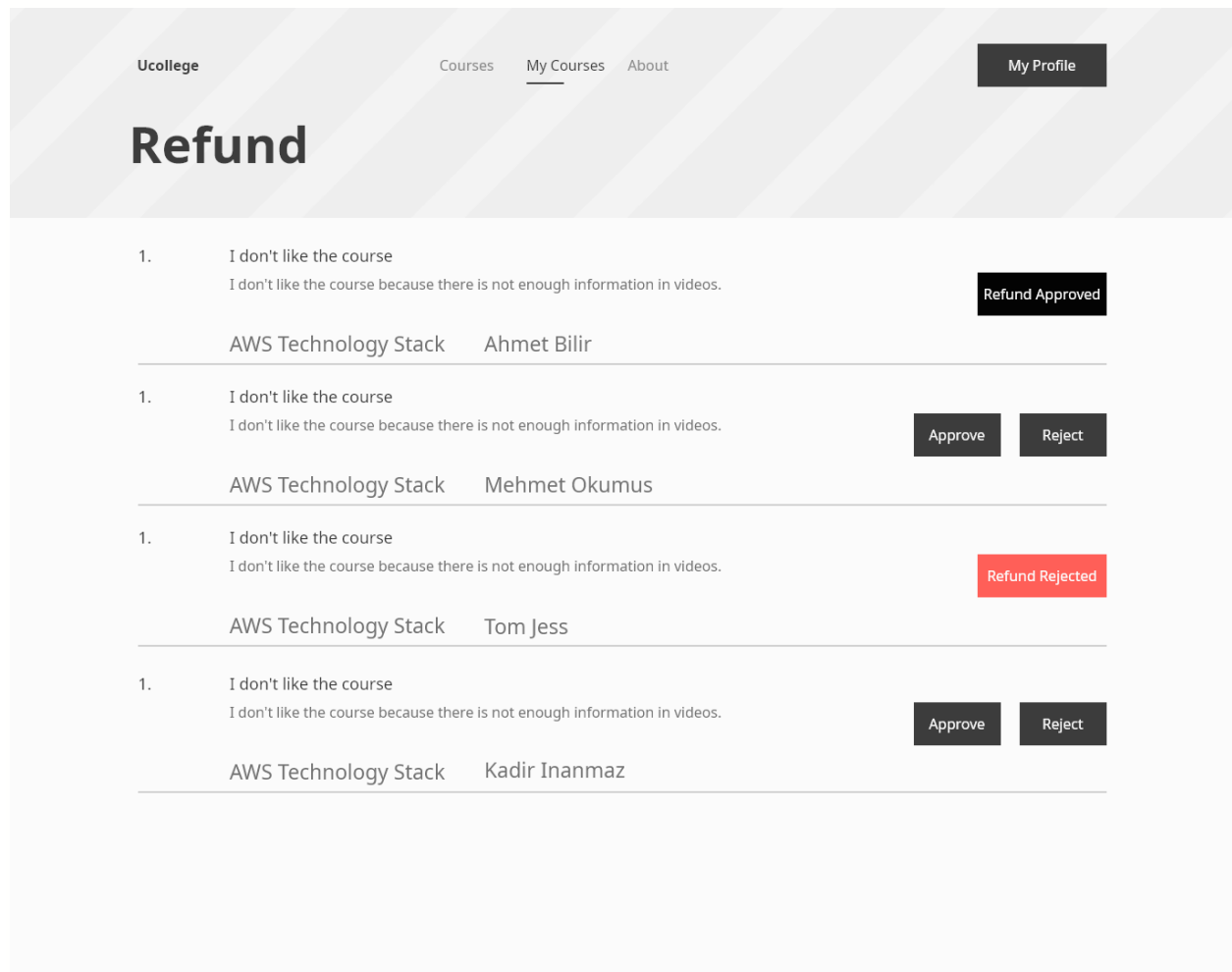
- Start Date:** A text input field with the value '2020-01-01' and a calendar icon.
- End Date:** A text input field with the value '2020-01-01' and a calendar icon.
- Calendar Pickers:** Two calendar pickers for June 2020. The first calendar has the 16th selected, and the second calendar also has the 16th selected.
- Percentage:** A text input field.
- Course ID:** A text input field.
- Create Discount:** A dark button at the bottom of the form.

- Create a discount for a specific course

**INSERT INTO** Discount(percentage, startDate, endDate, code, course\_id, admin\_id)

**VALUES** (<percentage>, <startDate>, <endDate>, <course\_id>, @admin\_id);

## Check Refunds Page



- List all refunds

**SELECT** \*

**FROM** Refund ref, Course c, Person p

**WHERE** ref.course\_id = c.ID **AND** ref.user\_id = p.ID **AND** p.ID = @user\_id;

- Update states

**UPDATE** Refund

**SET** state = <selected\_state>, admin\_id = @admin\_id

**WHERE** ID = @refund\_id;

# Implementation Plan

We are planning to use the Django framework for the backend and MySQL as the database. For UI, we will use HTML and CSS with some CSS libraries (Bootstrap, Tailwind etc.).

## Website

[CS353 Project Group 7 \(muratangin187.github.io\)](https://muratangin187.github.io)