

Project 1.2 COSI 127B

Due: Friday 1st March, 2024 at 23:59

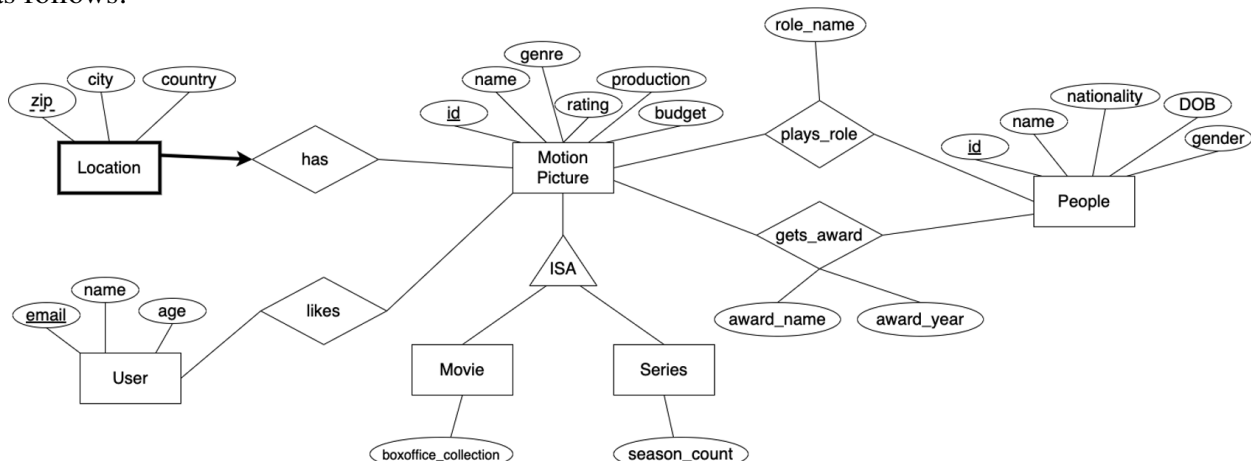
In the first project, you will design a simple website like IMDB that uses a movie database. This application will have a simple UI that is connected to a MySQL database.

In the first deliverable, you have already submitted an Entity-Relationship Diagram (ER Diagram) for the application, along with a relational schema. For this next deliverable, you will create a simple UI that connects to the back-end (i.e., database). We will provide you with the required schema description to be used for the rest of the project. For helpful links and setup, please look at the instructions 4.3.

We again recommend that you start as early as possible on this project.

1 Schema Description

An infotainment startup plans to create a website that will allow its users to view details of various motion pictures. Registered users will be able to view the list of all movies, search movies by name, genre, actor etc. Besides users will be able to 'like' movies. The required ER Diagram for the database is as follows:



Below, we list the tables required in the relational schema along with their primary and foreign keys:

MotionPicture (id, name, rating, production, budget)

User (email, name, age)

Likes (uemail, mpid)

Movie (mpid, boxoffice_collection)

Series (mpid, season_count)

People (id, name, nationality, dob, gender)

Role (mpid, pid, role_name)

Award (mpid, pid, award_name, award_year)

Genre (mpid, genre_name)

Location (mpid, zip, city, country)

Note: Primary keys are underlined and foreign keys are in **blue**.

2 Part 1: Building the Relational Schema

- (a) Build the given schema in a MySQL database in your local system using "CREATE TABLE" statements.
- (b) Pay attention to participation constraints, especially with weak entities. Also make sure to identify all foreign keys and primary keys (if required).

3 Part 2: Creating a UI

We need a simple UI to execute and display results to the users. Below, we list the basic requirements of the UI:

- (a) The user, through the UI should be able to execute a list of queries (we will list the required queries in the next deliverable) that should be parametrized when necessary. For this part (Project 1.2), just for sanity checking, implement few basic queries like – (i) keep a button ‘view all movies’; upon clicking it, it will list out all movies and their details, (ii) keep a button ‘view all actors’, which will work accordingly.
- (b) The queries can be executed from the front-end by clicking on links or buttons. Textbox, check boxes or option buttons can be used for parametrization requirements of queries.
Note: only providing a textbox where the user will manually enter every query is unacceptable.
- (c) The results of the queries should be displayed in a tabular format if the result has multiple tuples and columns.
- (d) For the requirement of “users liking movies”, the UI should support a way of requesting the user’s info and accept the user liking a movie through a button. Remember, the likes table must be updated if a user likes a movie.

These are a few basic requirements for the UI, and improvements to the front-end design to meet the required functionality is left to the student’s perspective. Please visit the office hours if you need any help.

4 Logistics

4.1 Collaboration

This is to be a group project with maximum size of 2.

4.2 Submitting your assignment

Please use the group submission option while submitting your assignment. Please place all the relevant php files (index.php and other files if you use multiple pages in your application) under a folder named Project 1.2. For this assignment we have two submission options:

1. Upload Project 1.2 folder to a git repository. Gradescope allows you to submit a link to a github or bitbucket repo so please use that option.
2. Zip Project 1.2 folder and submit the zipped file on gradescope. Make sure you discard all irrelevant files (like .Docstore) before clicking on upload.

The due date for this first part (PA 1.2) is Friday 1st March, 2024 at 23:59.

4.3 Helper Links

<https://getbootstrap.com/docs/4.0/components/forms/>

<https://www.w3schools.com/>

Instructions for setup.

Step 1: Download and install XAMPP.

Step 2: Open XAMPP and start all servers from Manage Servers tab.

Step 3: If all services turns to **green**.

Step 4: Open the browser; Open 127.0.0.1:80 and go to phpMyAdmin.

Step 5: Create a new Database and name it “COSI127b”.

Step 6: Under the COSI127b database, create a new table and call it “guests”.

(a) guests has 4 attributes - id, first_name, last_name, age.

Step 7: Insert at least 3 entries to this table.

Step 8: Download the index.php file from the course website. Here

Step 9: Look at the source code and check if you need to add a password (line 76-79).

Step 10: Find the installed location for XAMPP.

(a) For Mac users: using cmd+space search for “xamppfiles” and open in finder.

Usually, it goes to Applications → XAMPP → xamppfiles → htdocs.

(b) For windows users, the default location mostly is: C:/Program Files/XAMPP/htdocs.

Step 11: Create a new directory called COSI127b under htdocs/ and under that, create another subdirectory called pa1/.

Step 12: Place the index.php file in the pa1/ directory.

Step 13: Make sure that your apache web server is running.

Step 14: Open the web browser and go to localhost/COSI127b/pa1

Step 15: You should see an output like the one below with “guests” table data.

COSI 127b

Connecting Front-End to MySQL DB

Guests

Firstname	Lastname
John	John
Jane	Jane

Step 16: Complete the requirements given in Section 3.

Common Error on macOS:.

ERROR: “xampp-osx-8.2.0-0-installer” cannot be opened because the developer cannot be verified.

POSSIBLE SOLUTION - Do the following steps

- Search for “Privacy and Security”
- Here, scroll down to the “Security” section and you’ll see the blocked app. Click the “Open Anyway” option and continue with your installation.

Note that before doing this, you’ve already attempted installing the app and the error message has been thrown.

Common Error for Starting MySQL Database:.

ERROR: When you try to start the service it may not run

POSSIBLE SOLUTION - This can happen because port 3306 (default for MariaDB) is already in use.

- Check for port 3306.
For macOS / Linux - `sudo lsof -i :3306`
For windows - Look into services; right click and stop the service.
- If it shows some service (mostly it will be mariadb that we installed for WA 3), you have to stop it
For macOS - `brew services stop mariadb`
For Linux - `sudo systemctl stop mysqld`