

CS143 Projects

As part of class assignments, students will have to finish two class projects that are split into multiple submissions.

Prerequisite

We will assume that students are sufficiently proficient already with UNIX and have enough programming experience to learn PHP programming language quickly. We expect the resources and links on the project pages will provide you with enough information to get you started even in case you are not familiar with either one.

System Setup

To help students set up the uniform environment for the class project, we will be using [VirtualBox](#) to run the Linux operating system in a [virtual machine](#). VirtualBox allows a single machine to share resources and run multiple operating systems simultaneously. You will need to download the following files

- [VirtualBox binary file](#) for your host Operating System
- VirtualBox Image: [CS143.ova](#) - This is a very large file (~1GB), so it may take a while to download.

and follow our [VirtualBox setup instruction](#) to install VirtualBox on your own machine.

The provided virtual machine image is based on Ubuntu 14.04, MySQL 5.5.44, Apache 2.4.7, PHP 5.5.9, and g++ 4.8.4. If you have access to an equivalent machine that has MySQL, Apache2, PHP, and g++ installed, you may use it instead of the virtual machine image. However, please note that we cannot provide support for systems other than the virtual machine image, and that **your project MUST be runnable on the provided virtual machine**. We will be using the virtual machine image for grading purposes, and if your submission does not work within this setup, you may get zero points. We cannot make any exceptions to your project schedule for problems incurred by using your own computing facilities.

Project 1

Your project is to build a Web site on movies, actors and their reviews, supported by a relational database system. Your system manages all of its data at the back-end in a MySQL database and provides a Web interface to the users at the front-end. While this task may sound daunting at first, you will be surprised how easy and helpful a commercial DBMS is for such a project.

To develop Project 1 all students will use the MySQL DBMS and the Linux operating system. You will also use the PHP programming language to access MySQL and the Apache2 Web server to provide a Web interface. While there are many other ways to develop Web applications, Apache and PHP is one of the most popular method these days to implement a database-backed server for a small-to-medium scale Web site, so we want our students to learn how to do it. We will provide online references that will help you learn PHP and MySQL. We will also give you a considerable amount of real data to populate your system.

For Project 1, we will expect certain minimal functionality in your system - beyond that, the sky's the limit. Minimal functionality includes a variety of queries and browsing capabilities over the data in your system, and the capabilities to enter new data. Various integrity constraints must be monitored. Although you will use MySQL, which has reasonable transaction support, multi-user issues are not a focus of the project. Project 1 consists of two parts:

Part A: MySQL database

In this part, you will have to interact with the MySQL database to create tables, populate the database using our movie and actor data, add "integrity constraints" that the database should satisfy, and run a few SELECT queries. You will also learn how to connect PHP with MySQL, allowing a user to query the database through the website. A user of your interface should be able to execute SQL SELECT statements.

Part B: Movie database Web site

In this part, you will finally make user-friendly Web site for our movie database (a la IMDB.com), adding Web pages and functionality that are needed for database interaction. You will have to write a few PHP pages that execute various queries to find the information requested and to render the page in a user-friendly format. This is an opportunity for you to demonstrate some creativity and personal style. We may choose one or two outstanding projects for demonstration during one of the discussion sections.

Project 2

Apache Spark UDF Caching.

Partners

Students may implement the project individually or in teams of two. The choice is up to each student, but please keep the following rules in mind when you select your project partner:

An identical amount of work is expected and the same grading scale is used for individual and team projects. Faculty experience indicates that in general it is not necessarily easier or more productive to work in teams of two - it's largely a matter of personal preference and working style. If you choose to work as a team, you are encouraged to make use of collaborative authoring tools for synchronizing your work and ideas, such as version control software (e.g. [CVS](#), [SVN](#), [Perforce](#)) and online document tools (e.g. [Adobe Share](#), [Buzzword](#), [Google Docs](#)).

If you work in a team, choose your partner carefully. Teams are permitted to "divorce" at any time during the course (due to incompatibility, one partner dropping the course, or any other reason), and individual students may choose to team up as the project progresses, however students from divorced teams may not form new teams or join other teams. Put another way, if a student turns in any part of the project as part of a team, every later part of the project must be turned in individually or as part of the same team.

Both partners in a team will receive exactly the same grade for each project part turned in jointly. We will not entertain any complaints of the form "I did all the work and my partner did nothing." Choose your partner carefully!

If you work in a team, your work must be turned in jointly, as ONE submission. That is to say, only ONE of you two should submit your work as a team. Your team will get 10 points off as penalty if you violate this rule. Note that teamwork turned in as individual work will be considered as plagiarism and handled through official University channels.

Project References

Unix & VirtualBox

- [Unix tutorial](#)
- [VirtualBox overview](#)

PHP references

- [W3Schools PHP tutorial](#)
- [Input handling in PHP](#)
- [MySQL access in PHP](#)
- [PHP Regular Expression Tutorial](#)
- [PHP Perl Compatible Regular Expressions API](#)
- [PHP-MySQL API reference](#)
- [PHP function reference](#)

MySQL references

- [Introduction to MySQL](#)
- [Constraints in MySQL](#)
- [Accessing MySQL from PHP](#)
- [Date and Time types in MySQL](#)
- [MySQL documentation](#)
- [PHP-MySQL API reference](#)

C++ references

- [GNU C++ tutorial](#)
- [GNU Debugger tutorial](#)
- [Make tutorial](#)
- [C++ pointer tutorial](#)
- [C++ pointers vs references](#)
- [C++ const declaration](#)
- [C++ namespace](#)
- [C++ string](#)
- [C++ vector](#)