

## CS144 Project

### Project Overview

Your project is to build an online Web site that allows users to post blog entries written in markdown. Over the quarter, you will need to implement the Web site twice, once using a more “traditional” stack based on MySQL and Apache Tomcat, and once more using a “modern” stack based on MongoDB, Node.js, Express, and Angular (a.k.a. MEAN stack). Through this project, you will get familiar with the following software tools:

- Docker
- Apache Tomcat
- MongoDB
- Node.js
- Express
- Angular

In addition to the above software tools, you will gain experience with Servlet programming in Java, and building Web applications using HTML, CSS, and Javascript.

The project should be done either individually or by a team of two students.

### System and Programming Issues

In order to minimize the installation and configuration hassle, we will provide “Docker container images” that include all necessary software packages needed for the project. All your development should be done inside a “virtual environment” (a.k.a. *container*) created by our image. In order to run the provided container, you will need a fairly-equipped machine, with at least 4GB of main memory and 10GB of free hard disk space.

The first two projects will be done primarily in Java on Apache Tomcat, while later projects will be done in Javascript on the MEAN stack. Even now, Java is one of the most popular methods used in the enterprise setting for server-side projects, so we want our students to be familiar with the language. Then in later projects, students will

primarily develop in Javascript on the MEAN stack, a Web development stack that have become hugely popular in the last few years.

We will assume that students are already proficient with SQL and Java or able to learn them quickly. For those of you who haven't done much Java programming before, please note that the syntax and the programming model of Java is very similar to C++. As long as you are a proficient C++ programmer, even without much prior Java experience, we expect that that you can finish our first projects in a reasonable amount of time. We do **NOT** assume any experience in Javascript programming. Later projects that are based on Javascript will be done only after we learn Javascript in the class.

## Project Parts

Tentatively, we plan to break the project in five parts:

### Part 1: System Setup and Warm-up

In this part, you will have to set up a provided "container" on your computer and interact with MySQL to populate data and query them. You will also have to implement a simple Java program to brush up on Java programming.

### Part 2: Markdown Editor on Tomcat

In this part, you will need to implement a simple online markdown editor using MySQL and Tomcat.

### Part 3: Markdown Editor Using Angular

In this part, you will use a more advanced and dynamic version of markdown editor client using Angular. All entries saved by the editor will be stored *locally within the browser*, without any interaction with the server yet.

## Part 4: Blogging Server

Now we will write the server-side program that saves and publishes blog posts written by the markdown-editor client built in Part 3. The server-side program will be developed using Node.js and Express. All data will be stored in the server side using MongoDB.

## Part 5: Performance and Scalability

In the last part of our project, you will explore issues of performance and scalability of Web applications.

## Late Submission Policy

To accommodate the emergencies that a student may encounter, each student (or team) has a 4-day grace period for late submission to use throughout the quarter. The grace period can be used for any part of the project in the unit of one day. For example, a student may use 1-day grace period for part 1 and 2-day grace period for part 3. **Any single project part may not be more than 2 days late.** Note that the grace period can be used in the unit of one day. even if a student submits a project 12 hours late, he/she needs to use a full day grace period to avoid late penalty.

Once you run out of your four day grace period, you have to contact course staff **in advance** to get an approval for having a “special-case” grace period. Otherwise, you will get a ZERO score for the delayed submission. For fairness, after getting the approval, you will still get an (accumulative) additional 20% off per day for using the “special-case” grace period, i.e. 20% off if using 5th day grace period, 40% off if using 6th day, etc. Again, any single project part may not be more than 2 days late.

## Partners

Project 1 should be completely and submitted by every student, but starting from Project 2, 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. Git, SVN, Perforce) and online document tools (e.g. Google Drive).

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! Also, if both students already submitted their Project X separately, they cannot form a team in the Project X. In other words, they are considered as two individual submissions in the Project X. Thus, any type of collaboration is still prohibited. If you might (or plan to) form a team in a project, do NOT submit your work to our system unless you made your mind.

If you work in a team, your work must be turned in jointly, as **ONE** submission. If we receive submissions from students in a team separately, they would get 10% off as a penalty. In addition, if the above separated submissions are different, the submission with a lower score prevails. Teamwork turned in as individual work will be considered as plagiarism and handled through official University channels.

### Academic Honesty

Each of you is expected to submit your own original work, or the original work of your two-student team in the project. On many occasions it is useful to ask others (the instructor, the TA, or other students) for hints or debugging help, or to talk generally about programming strategies. Such activity is both acceptable and encouraged, but you must indicate any assistance (human or otherwise) that you received. Any assistance

received that is not given proper citation will be considered plagiarism. In any event, you are responsible for coding, understanding, and being able to explain on your own or as a team all project work that you submit.

We will pursue aggressively all suspected cases of plagiarism, and they will be handled through official University channels.

If you have any questions about this policy or about the degree to which we will pursue academic honesty violations, please discuss your concerns with the course staff immediately.