



CSI 44 Web Applications Discussion Session

Week- I

Logistics

- Instructor: Junghoo (John) Cho
 - Office Hour: Tue. 2:30PM - 3:30PM, BH 353 I H
- TAs: Jiaqi Gu, Zijun Xue, Jin Wang, Zhehan Li
 - Office Hour: Wed 10 - 12 PM, Fri 12 - 2 PM, BH355 I
- Discussion Sessions on Friday
 - We will hold discussions every week unless special notices.
- TAs take turns to hold discussion sessions.
 - May not be the same TA teaching as listed on CCLE, but all four sessions are available every week.
 - Same materials are covered in any of the four session.
 - You choose to attend **any** one, regardless of which one you've enrolled in.

Online Discussion Forum (Piazza)

- Piazza (Strongly Recommended)
 - <https://piazza.com/class/jc5ebaeewf534x>
 - Lecture/Project related Q&A – **Piazza Posts**
 - Grades/Personal – **Piazza Private Note**
 - Only you and the course staff can see private post
 - Do not post general questions in private
 - Please make them public so other students can benefit from the answer
- Emails (Not Recommended)

Piazza questions

- We typically not meet in person one on one
 - Make your questions clear, specific and in details
- Project related questions
 - If you encounter any issues, i.e. a bug, please think and debug it by yourself first before posting the question.
 - Projects are designed for you to know how to BE PATIENT, DILIGENT, and SELF-LEARNER
 - TAs typically will not help debug your code.
 - It's an upper-level CS course and TAs won't help much as in an entry-level CS course (i.e. CS31, 32)

Academic Honesty

- Do not risk anything related to **plagiarism!**
 - No copy code from others
 - No cheating in exams
- Take time to do the project by yourself
 - Please do not disclose your answer (in Piazza or any other website, i.e. Github)
 - When you ask project-related questions, please take care that you do not post your project's code
 - A workaround: rewrite it in a way that your answer is not revealed.

Grading

- Final Exam - 40%
- Projects (5 parts) - 60%
- Note that project counts 60%. The final grading will be done based on the curve. Roughly 30% students will get A, 40% B and the remaining 30% C or D

Teamwork in Projects

- Starting from Project 2, students can work in a team of two to complete later projects.
- Team Divorce Policy
 - Team up only once!
 - 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.

What if two students become a team?

- Only one copy needs to be submitted per team:

Please have your submission always uploaded by the same person in your team in every project.

- Both team members will get the same score and are able to check it on CCLE.

Projects

- Projects are very important in this course.
- It is necessary to read **every word** in each project's description page.
 - In our experience, most of the mistakes come from ignorance of part of the project description.
 - Especially those highlighted words.



When you submit the project...

Rule 1:

Make sure it can run !!!

TEST on a CLEAN environment
(the docker image we provided you)
before your submission is always a
good idea.

What if it can't run?

- You will get 0 points.

Rule 2: Grace Period

- Use it wisely!!
- At most 2 grace days per Project.
 - Total 4 days of all projects (per team)
 - If you (your team) used up 4 days, **must** email Professor for additional days (or get 0 points!)
 - Even if permitted, you will still get 20% penalty for each late day
 - i.e. 5th day 20% off, 6th day 40% off, ...



Rule 3:

**You can submit more than once.
CCLE will record your last
submission date and time (Do not
re-submit after the deadline!)**

Projects Overview

- An online Blogging web site
 - Post and view blogs in markdown
- Implement site twice over the quarter
 - Use a more "traditional" stack
 - Java, MySQL, Apache Tomcat
 - Use a more "modern" stack
 - MongoDB, Node.js, Express, and Angular
 - MEAN stack
- Many funny and useful technologies to learn during the quarter!

Project Parts

- Part 1: System Setup and Warm-up
- Part 2: Markdown Editor on Tomcat
 - "traditional" stack
- Part 3: Markdown Editor Using Angular
 - "modern" stack – frontend
- Part 4: Blogging Server
 - "modern" stack – backend
- Part 5: Performance and Scalability
 - explore challenging issues faced by many web sites today

Project Docker Image

- Docker Setup:
 - <http://oak.cs.ucla.edu/classes/cs144/docker/index.html>
 - The Docker image for Project 1 (and Project 2), "junghoo/cs144-tomcat", has MySQL, Apache Tomcat, and Java JDK 8 installed.
- Make sure you follow instructions on the guide to avoid common issues.

Project I Overview

- Setup Docker on your machine
- Get familiar with Unix CLI
- Write actors.sql to load and process data
- Write a Java code to compute SHA-1
 - sha1sum
 - Fully test on different **text** and **binary** files!
- Create a submission zip and submit through CCLE
 - No email submissions are accepted.

Week 1 Brief Review

- How Web Browsers and Web Servers Communicate?
 - TCP/IP
 - transmission control protocol and internet protocol
 - DNS (domain name service)
 - internet protocol to map domain names to IPs
 - HTTP (hypertext transportation protocol)
 - application protocol on top of TCP/IP
 - communication protocol between web servers and web clients
 - HTML (hypertext markup language)
 - page formatting and linking standard

HTTP

- HTTP/1.1 most popular (HTTP/2 is the most recent)
 - Request & Response
 - all interactions start with a client's request
 - Stateless
 - every request is handled independently from others
 - Request Message = request line + header + body
 - Response Message = status line+ header + body
- Response Status:
 - 2XX: Success
 - 3XX: Redirection
 - 4XX: Client Error
 - 5XX: Server Error

HTML

- HTML 4.01, HTML 5
- Document = text + tags
 - HTML Tags are enclosed in `<...>`
 - Comments appear in `<!-- ... -->`
 - Embed a link `...`
 - Overall structure

```
<!DOCTYPE html>
<html>
<head>...</head>
<body>...</body>
</html>
```

HTML Forms

- `<form action="http://www.google.com/search" method="GET">`
 `<input name="q" type="text"><input type="submit">`
 `</form>`
- Method:
 - GET: input values are encoded within URL
 e.g. <http://www.google.com/search?q=yahoo>
 - POST: input values are encoded in the body of the request
- Post Method Example:
 - <http://oak.cs.ucla.edu/classes/cs144/examples/post.html>