

CS 142 Final Examination

Spring Quarter 2019

You have 3 hours (180 minutes) for this examination; the number of points for each question indicates roughly how many minutes you should spend on that question. Make sure you print your name and sign the Honor Code below. During the examination you may consult two double-sided pages of notes; all other sources of information, including laptops, cell phones, etc. are prohibited.

I acknowledge and accept the Stanford University Honor Code. I have neither given nor received aid in answering the questions on this examination.

(Signature)

(Print your name, legibly!)

_____@stanford.edu
(SUID - Stanford email account for grading database key)

Problem	#1	#2	#3	#4	#5	#6	#7	#8	#9	
Points	12	12	10	10	8	8	8	8	8	
Problem	#10	#11	#12	#13	#14	#15	#16	#17	#18	Total
Points	8	10	12	10	12	10	10	12	12	180

Problem #1 (12 points)

(a) Assume you have joined a new international standards body for the Internet that has decided to take on the problem of misleading names. The standards body has taken up a complaint against the names *HyperText Transport Protocol* (HTTP) and the DOM's *XMLHttpRequest*. Using our class's Photo App as an example, describe what is misleading about the name and why it is misleading for our Photo App:

A. The word "HyperText" in HyperText Transport Protocol.

B. The acronym XML in XMLHttpRequest.

Problem #2 (12 points)

- (a) If you display a web page in the Chrome browser that is entirely fetched using the HTTPS protocol, Chrome will display  next to the URL. If you change the web page to include a small image of something public (e.g. a logo) and use HTTP to fetch the image Chrome will display  next to the URL as if you didn't use HTTPS for anything. Explain why the Chrome developers could justify slapping "Not Secure" on something seemingly benign as an HTTP fetch of a small image. Describe the attack they are worried about.

- (b) Your friend has started a company with a photo sharing web app. In order to associate each request to the server with a logged in user, she includes the user ID in the cookie. On each incoming request, the server checks that field in the cookie to figure out which user the request is coming from. (1) What is problematic about this? (2) What is the correct way to achieve the same goal? (3) Would switching to use HTTPS solve the problem?

Problem #5 (8 points)

A student asked the following question on Piazza:

In the projects, how is the photo-share.html file loaded? In the projects 5-8, how is the photo-share.html loaded from the server to the browser. It seems that all the functions in the webServer.js (app.get() or app.post()) are only loading the model data. Does loading photo-share.html even use webServer.js? Thank you.

Write a good answer to the question. In the spirit of the course theme of how web apps work, your answer should include a description of the mechanism that gets the photo-share.html file into the browser.

Problem #6 (8 points)

Web app developers use the term "Server Push" to describe having a Web Server with the ability to update information in Browsers by "pushing" it from the Web Server to Browser. Describe something fundamental about the HTTP protocol that makes it less than ideal for implementing "Server Push".

Problem #7 (8 points)

In class we learned that Memcache is a storage system that is used to store session state and cache results of database queries. It is used because it is significantly faster than accessing data stored in database. Explain why we don't use Memcache for all our web app's storage needs.

Problem #8 (8 points)

Explain why it is easier to apply a scale-out approach to web servers than database servers.

Problem #9 (8 points)

Describe the key property of cloud computing services (e.g. Amazon Web Services) that make it a game changer for web applications that face uncertain but possibly explosive growth when compared to owning your own servers.

Problem #10 (8 points)

Assume a new version of the Chrome came out that contained a bug that broke enforcement of the same origin policy. Accesses that were previously prevented by the same origin policy are now permitted. Explain what an attacker running in a different browser tab could do to our Photo App. Describe the attack.

Problem #11 (10 points)

(a) Describe a Denial of Service(DoS) attack that would work on your Project #7.

(b) Describe how you could change your Project #7 to mitigate the attack.

Problem #12 (12 points)

Consider the following Node.js program:

```
var async = require('async');

var arr = ["A", "B"];
var printOrder = [];

async.each(arr, function (elem1, doneCallback) {
  async.each(arr, function (elem2, doneCallback2) {
    console.log(elem1 + elem2);
    printOrder.push(elem1 + elem2);
    doneCallback2();
  }, function () {
    console.log('innerDone', elem1);
    doneCallback();
  });
}, function() {
  console.log('outerDone');
  console.log("P01", printOrder);
})
console.log("P02", printOrder);
```

(a) Execute the code by hand and show a possible output from the program. (Hint: (elem1 + elem2) concatenates the two characters)

(b) Is it possible that the console.log lines that do not contain the string "Done" can occur in a different order on different runs? Justify your answer

Problem #13 (10 points)

(a) Explain how Cross-Site Request Forgery (CSRF) attack can allow a website running in a different browser tab to perform operations on a web app like it is logged into the web app even though the attacker doesn't know the user name or password of the user in the web app.

(b) Explain why the browser can not just turn off the mechanism used by CSRF. Give an example of something useful that breaks with the mechanism turned off.

Problem #14 (12 points)

(a) Database systems that support primary and secondary indexes will allow a user to have multiple secondary indexes but only one primary index on a table. Explain the reason for this.

(b) When an index is added to a collection in MongoDB, it could result in a performance increase or a performance decrease, describe how you could look at a system running and predict if adding an index would help or hurt performance. Give examples that show both cases of helping and hurting.

Problem #15 (10 points)

You have been appointed a CA for CS142 and are approached by a student during office hours for a bug in Project #6. Instead of using `axios` to talk to the web server, they used another JavaScript library called `getIt` that had similar interface to `axios.get`. The student describe the following.

- The student tried doing:

```
let userList = getIt("/usr/list");
```

and discovered `userList` was an object but didn't contain the model data properties from `/user/list`.

- The student then discovered that if they did:

```
let userList = getIt("/usr/list");
setTimeout(function(){ processModel(userList. fulfillmentValue);}, 10);
```

the system worked. Although the `userList` never got set to the model data some property named `fulfillmentValue` ended up having the model data even though it wasn't set immediately after the call to `getIt`.

- (a) Write an explanation to the student what is going on with their code.

- (b) Show the code you would suggest to properly use `getIt`. Your solution should pass the response data to the routine `processModel` as was done in the code above.

```
let userList = getIt("/usr/list");
```

Problem #16 (10 points)

(a) The relational data model stores data in tables consisting of columns and rows. Assume you have been given the task of moving data in an object database like MongoDB. Describe how each of the following concepts from a relational database would map to something in the object model:

(i) A Table

(ii) A Row in a Table

(iii) A Column in a Table

(iv) A value in a particular Row and Column in a Table

(b) It is possible to map a relational data model into an object model. Explain why the reverse, mapping an object model (i.e. JSON-like MongoDB) to a relational database, doesn't work.

Problem #17 (12 points)

(a) When doing routing of URLs in both React Router and in ExpressJS in our photo app, we ended up having routes that contained a colon character (e.g. `"/foo/:bar"`) yet we never included a colon in the hierarchical part of a URL we used.

(i) Explain the purpose of this colon character?

(ii) Describe what would happen if we just deleted the colon from the routes.

(b) Our use of ExpressJS in the Photo App took advantage of much functionality offer as Express Middleware. If we didn't use the Middleware mechanism, could we have implemented the same functionality in our `webServer.js`?

If your answer is no, describe the functionality we couldn't achieve without Middleware.

If your answer is yes, explain the disadvantage of not using Middleware.

Problem #18 (12 points)

(a) In your photo app assignment, you were encouraged to validate user input in both your back-end server and your front-end React application. However, some errors can only be handled on the back-end; give an example of such an error and explain why it cannot be validated on the front-end.

(b) Our photo app assignment accessed the MongoDB database using the Mongoose object definition language rather than talking directly to MongoDB. Describe the advantages this setup of going through Mongoose gave us when compared to accessing MongoDB directly.