請勿攜去

香 港 中 文 大 學

The Chinese University of Hong Kong

二 0 一九至二 0 年度暑期課程科目考試

Course Examination Summer Term, 2019-20

科目編號及名稱
Course Code & Title : CSCI2720 – Building Web Applications

時間 小時 分鐘
Time allowed : 1 hours 20 minutes

學號 座號
Student I.D. No. : Seat No. :

Instructions

- This is a closed-book examination, with one double-sided A4 written cheat sheet allowed
- No electronic device is allowed
- Write your answers clearly in the space provided in this exam script
- You need to return this exam script with all pages for grading
- Assume that the web browser in used is the almost-latest version of Google Chrome, and all libraries and frameworks are of the version used in the course
- Cross out all draft or rough work if they are not to be graded
- Please keep your answers legible, precise and concise

Markers' use only

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	**Total**
/13	/10	/5	/10	/12	/18	/20	/12	**/100**

Question 1: HTTP (13%)

a) Comparing to POST, what are the effects of GET not storing data in the body? Briefly describe two characteristics. (6%)

b) Here you can see an HTTP request and its corresponding response: (7%)

```
GET https://www.cse.cuhk.edu.hk/en/
Host: www.cse.cuhk.edu.hk
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:72.0) Gecko/20100101
Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK                    line "A"
Date: Thu, 16 Jul 2020 09:12:52 GMT
Server: Apache
Expires: Wed, 17 Aug 2005 00:00:00 GMT
Last-Modified: Thu, 16 Jul 2020 09:12:52 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Set-Cookie: ad42a4851192f3729c26ee68fa291805=c70c06502d0fe759bd6e11d954c42a22;
path=/; HttpOnly                                                    line "B"
Connection: close
Transfer-Encoding: chunked
...
```

1. What can you conclude about the software of the client and server respectively?

2. What can you conclude from *line "A"*?

3. What can you conclude from *line "B"*?

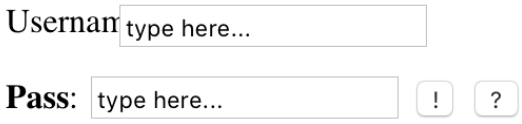
Question 2: RESTful API (10%)

a) Why does it mean, when we say that a GET request without query string is always *idempotent*? (4%)

b) Why is RESTful API useful? (6%)

Question 3: UI/UX Design (5%)

What are UI and UX design respectively? Please give an example with reason on how UI and UX each can be improved, basing on this layout:



Question 4: React (10%)

The React JSX code for an app in **`div#app`** is given as below:

```
const data = [
  {filename: "cuhk-2013.jpg", year: 2013, remarks: "Sunset over CUHK"},
  {filename: "cuhk-2017.jpg", year: 2017, remarks: "Bird's-eye view of CUHK"},
  {filename: "sci-2013.jpg", year: 2013, remarks: "The CUHK Emblem"},
  {filename: "shb-2013.jpg", year: 2013, remarks: "The Engineering Buildings"},
  {filename: "stream-2009.jpg", year: 2009, remarks: "Nature hidden in the campus"},
];

class App extends React.Component {
  render() { return (
      <>
        <Header />
        <FileCard />
      </>
    )}
}

class Header extends React.Component {
  render() { return (
      <header className="bg-warning">
        <h1 className="display-4 text-center">{this.props.name}</h1>
      </header>
    )}
}

class FileCard extends React.Component {
  constructor(props) {
    super(props);
    this.state = {selected: -1};
  }
  handleClick(index, e) {
    if (this.state.selected != index)
      this.state = {selected: index};
    else
      this.state = {selected: -1};
  }
  render() { return (
      <main class="container">
      { data.map((file,index) => (
         <div key={index} className="card d-inline-block m-2" onClick={(e) =>
this.handleClick(index, e)} style={{ width: this.state.selected==index ? '100%' :
200 }}>
           <img src={"images/"+file.name} alt="{file.remarks}" className="w-100" />
           <div className="card-body">
             <h6 className="card-title">{file.name}</h6>
             <p className="card-text">Year: {file.year}</p>
             { this.state.selected==index &&
               <p className="card-text">{file.remarks}</p>
             }
           </div>
         </div>
      ))}
      </main>
    )}
}

ReactDOM.render(<App name="CUHK Pictures" />, document.querySelector("#app"));
```

You might find this quite similar to Lab 5 you have worked on. Unfortunately, there are some issues to be fixed. Please name the line(s) to be modified in each case, and write down the correct code for the line(s).

a) Symptom 1: *The header is not shown*

b) Symptom 2: *The change of state "selected" is not successful*

c) Symptom 3: *The images are not loaded*

Question 5: Serverless Computing (12%)

a) What is FaaS? Why is it getting popular recently? What are its drawbacks?
Give two examples on each. (8%)

Function as Service

b) What kind of service does AWS Lambda provide? Give an example on how it can be useful with Node.js. (4%)

Question 6: Security (18%)

a) How can an API endpoint be protected from security issues? Name two possible ways. (2%)

b) Why should a web application validate user input? Elaborate on two possible threats of using user input directly into the app logic without processing. (8%)

c) Give a pair of examples each, on two different ways URLs would be considered cross-origin, with hostnames being `*.cuhk.edu.hk`. Name one threat which could arise with badly configured CORS policies and explain briefly. (8%)

Question 7: Express and Mongoose (20%)

For a web site serving data of mobility gadgets, you need to store the following data:

- Information of *gadgets*, which should include the gadget **name**, the **year**, the **brand** (one *company*), and a **ranking** (e.g. 4.9)
- Information of *companies*, which should include the **name**, and the year **established**

In this database, only the names cannot be missing.

a) Please design efficient schemas for these data to be used in Mongoose. Put your assumptions in code comments if necessary. (6%)

```
let express = require('express');
let app = express();

let mongoose = require('mongoose');
let Schema = mongoose.Schema;

mongoose.connect('          '); // assume that DB connects fine
let db = mongoose.connection;
db.on('open', function() {
  console.log("db ok");
});









let Gadget = mongoose.model('Gadget', gadgetSchema);
let Company = mongoose.model('Company', companySchema);

app.listen(2000);
```

b) The Express web server is used to accept requests for queries to the database. Write down the route based on Express **app** continuing on the code above (e.g. ***app.xxx()***) to deliver the raw query results (in JSON) as response to the following HTTP requests. (14%)

1. To look for the companies which has a year established earlier than *y*:

```
GET /company/year/y
```

2. To look for the one gadget with top or bottom ranking

```
GET /gadget?ranking=top      OR
GET /gadget?ranking=bottom
```

Question 8: SPA and History (12%)

a) What are the drawbacks of building a Single-Page App (SPA)? (6%)

b) 1. Describe what would happen, if this is executed in the JavaScript console when your current page is `http://www.cuhk.edu.hk` in a browser: (4%)

```
window.history.pushState({url:'cse'}, "/cse", "/cse");
```

2. What would happen if the user clicks on the Back button now? (2%)

~ END OF THE EXAM ~