Final Exam Practice Questions

CSCI UA.0480-002 – Applied Internet Technology

1. What are some **options** (**name at least 3**) for **storing data** for your web application. **Discuss why you would or would not use** that option to store your data.

2. Draw lines to match the http request with the corresponding **CRUD** operation:

GET	Create
POST	Delete
PUT	Update
DELETE	Read

3. Answer the following questions about using MongoDB:

   a) How do you display all **databases** in the MongoDB shell?

   b) How do you display all **collections** in the MongoDB shell?

   c) What's a **collection**?

   d) Given the following movie related documents in a database called `moviedb` and collection called `movies`, what query would you use in the MongoDB shell to **retrieve all** movies?

```
[
    {'title':'Mauvais Sang', 'slug': 'mauvais-sang', 'year':1986},
    {'title':'Stroszek', 'slug': 'stroszek','year':1977},
    {'title':'Blue Velvet', 'slug': 'blue-velvet','year':1986},
    {'title':'Only Lovers Left Alive', 'slug':'only-lovers-left-alive', 'year':2013}
]
```

   e) Using the database/collection/documents from part d, what query would you use in the MongoDB shell to **retrieve all movies made in 1986**?

   f) Using the same database/collection/documents from the previous 2 questions, how would you **remove all of the movies**?

4. When generating an Express project, a project layout is automatically built for you. Describe what folders and files are created and what their significance is. Name and describe at least 4 files / directories.

5. What are the general steps for integrating Mongoose into your Express app?

6. Create a **Mongoose Schema** (or Schemas) to keep track of **contacts and their contact information**. It should support:

   a) storing a contact's first and last name, email, phone number and address
   b) each contact can have multiple addresses
   c) every address should contain a street address, city, state and zip code

   Write your Schema below. Although syntax doesn't have to be exactly correct, make an effort to write something that's reasonably close to what the actual code may look like.

7. List at least two best practices for saving passwords in a database. Describe the rationale behind each best practice.

8. Using the sample movie database, collection and documents from question #3, write the code that would go into your router, `index.js`:

   a) Assume that you have a corresponding Mongoose Schema that matches the fields of the documents from question 3 (**the slug will be handled automatically** by a url slug plugin, so you will not have to worry about that in your form or router)
   b) Additionally, a **model** related to the schema has been registered under the **name `Movie`**
   c) Imagine that you have the form markup listed at the end of this question
   d) In your router, `index.js`...
   e) (Remember to retrieving the model constructor, `Movie`)
   f) Write **two route handlers**...
   g) Create a route handler that accepts a **POST** from the form at the end of this question (determine the url to use from the form)
      - it should create and save a new movie, and redirect you to the movie detail page of the movie just created
      - the movie detail page will be at the url: `/movie/[movie-slug]` (for example `/movie/blue-velvet`)
   h) Add another route that handles a GET to the url : `/movie/[movie-slug]`
      - this route handler should retrieve the movie identified by the slug in the url from the database
      - ...and it should send over the movie's information to a template called movie-detail

```
<form method="POST"  action="/movie">
  <input type="text" name="title">
  <input type="text" name="year">
</form>
```

In `router.js,` you'll have some of the following setup code:

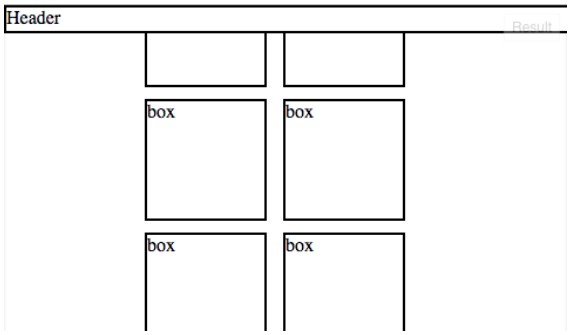
```
var express = require('express');
var router = express.Router();
var mongoose = require('mongoose');
```

Assume route handlers to get the form, etc ... are present in this file. Your code goes below (it does not have to be exactly syntactically correct, but it should be *reasonably* close):

9. In the previous question where your web application receives input from a user:

   a) describe where in your application you can add validation
   b) specify where ***must*** you have validation and include why

10. Create the markup and CSS for the following layout below:

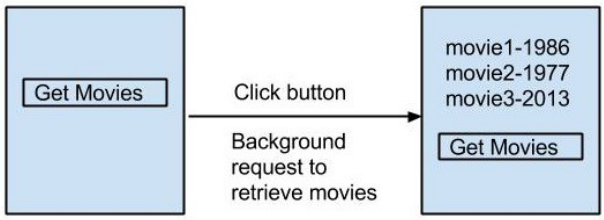


a) When the page is scrolled, the header does not move (it sticks to the top)
b) There is a grid of boxes on the page
c) The pixel dimensions do not have to be exact
d) The number of columns does not have to match exactly

Markup	**CSS**

11. **Write client side markup and code to**:

a) display a page with one button
b) when the button is clicked, send a background request (that is, no page reload; your JavaScript should send a request *behind the scenes*)
c) the request will got an api; the api is located at localhost:3000/api/movies (assume that the domain is the same domain that your client side script originates from)
d) the response from the api will be a JSON document that is exactly the same as the list of documents shown in #3, part d
e) once it successfully retrieves the document, insert every movie into the page's body as paragraph tags
f) each paragraph tags will contain the title and year of a move separated by a dash



g) the code does not have to be exact syntactically, but *reasonably* close

12. Write CSS selectors to pick out the specified elements from the markup below.

```
<h1>Header</h1>
<div>Hello!</div>
<div class="content">
    <a>Link 1</a>
    <p>
        <a>Link 2</a>
    </p>
</div>
<a>Link 3</a>
<a data-priority>Link 4</a>
```

a) `All a tags` _____

b) `All the a tags nested under the div with class content` _____

c) `Only the a tags that are the direct descendent of the div with class content` _____

d) `Both h1 tags and div tags in one selector` _____

e) `(Extra) Only the a tags with an attribute of data-priority` _____

13. List three methods/functions that allow you to pick out / obtain HTML Elements

14. When and where should client side JavaScript be loaded? Why?

15. Using the following CSS and markup, animate the div with id, animateMe, so that it travels across the page from left to right.

```
<div id="animateMe"></div>
```

```
#animateMe {
    position:absolute;
    width: 50px;
    height: 50px;
    background-color: #0f0;
    top:0px;
    left:0px;
}
```

16. Describe 3 possible values for the CSS display property... and what effect they have on layout.