

# **ECE 413/513 – Fall 2019**

## **Exam 2**

Monday, October 28, 2019

**Name:** \_\_\_\_\_

No booklets, additional paper, textbooks, or other materials may be referenced during this examination with the exception of one 8.5"x11" sheet of paper (front and back). Your only other resources are a pen/pencil, eraser, and this exam.

Submit answers only in space indicated on each page.

## 1. Multiple Choice Questions (2.5 points each)

(1.1) Which of the following jQuery statements gets the value of an input text element with ID `userRating`?

- (a) `$(userRating).value();`
- (b) `$('.userRating').text();`
- (c) `$('#userRating').val();`
- (d) `$('#userRating').text();`
- (e) None of the above

(1.2) What status code should be returned for an authentication failure of POST request for a RESTful endpoint?

- (a) 200
- (b) 201
- (c) 400
- (d) 401
- (e) None of the above

(1.3) Which HTTP request verb such be used for retrieving a resource?

- (a) POST
- (b) GET
- (c) PUT
- (d) DELETE
- (e) None of the above

(1.4) Which Mongoose query will return all Reviews with a rating greater than or equal to 4.0 sorted in descending order by review date.

- (a) `Reviews.find({"rating": {$ge:4.0}}).sort()`
- (b) `Reviews.find({"rating": {$gte:4.0}}).sort({"reviewDate"})`
- (c) `Reviews.find({"rating": {$gte:4.0}}).sort({"reviewDate": 'desc'})`
- (d) `Reviews.findAll({"rating": {$ge:4.0}}).sort({"reviewDate ": -1})`
- (e) None of the above

(1.5) `bcrypt.hash(password, 10, function(err, hash) {...});` does which of the following?

- (a) Encrypts `password` using a secret key of 10, passing the hashed password to the anonymous function.
- (b) Creates a cryptographic hash of `password` using salt, passing the hashed password to the anonymous function.
- (c) Creates and returned a cryptographic hash of `password` using salt.
- (d) Checks if the encrypted password `password` matches the value 10.
- (e) None of the above

**(1.6)** Which of the following determines if the JSON sent in the body of POST request contains the property `rating`?

- (a) `if (req.query.rating !== undefined) {`
- (b) `if (req.body.rating.hasOwnProperty()) {`
- (c) `if (req.body.hasOwnProperty("rating")) {`
- (d) `if (req.params.rating.hasOwnProperty()) {`
- (e) None of the above

**(1.7)** For an Express route, which of the following sends a 201 HTTP response code to the client?

- (a) `res.send(201);`
- (b) `res.send(201, JSON.stringify(responseJson));`
- (c) `res.send(JSON.stringify(responseJson), 201);`
- (d) `res.status(201).send(JSON.stringify(responseJson));`
- (e) None of the above

**(1.8)** Which determines if an authorization token named `authToken` is stored in local storage.

- (a) `if (localStorage.getItem("authToken")) {`
- (b) `if (window.getItem("authToken")) {`
- (c) `if (document.getItem("authToken")) {`
- (d) `if (window.localStorage.getItem("authToken")) {`
- (e) None of the above

## 2. Ajax jQuery (7.5 points)

Using jQuery's `$.ajax()` method and jQuery's DOM manipulation methods, write JavaScript to:

1. Make a POST request to the endpoint `/reviews/add` that passes a review object consisting of properties for `rating`, `comments`, and `date`. The API will return JSON with the new average rating.
  - a. The web page contains an `<input type="text">` element with ID `userRating` and a `<textarea>` element `userComments`, that are used to specify the `rating` and `comments` properties, respectively.
  - b. The current date should be used for the `date` property.
  - c. Uses anonymous functions for the success and failure response handlers
2. On success:
  - a. The endpoint will return JSON with the property `averageRating`.
  - b. The response handler (implemented using an anonymous function) update a `<input type="text">` with the ID `averageRating` with the new average rating.
3. On failure:
  - a. The response handler should update and show a `<div>` with the ID `messageDiv` to display the message `"Error contacting server."`

### 3. MongoDB and Mongoose (7.5 points)

Complete the schema definition for a Mongoose schema named `reviewSchema` containing the following:

1. `email`: Reviewer email, which is required and must be unique.
2. `ratings`: Ratings object consisting of separate ratings for food, service, and ambience.
3. `comments`: Reviewer comments.

Write your answer in the box provided.

```
var db = require("../db");
```

```
var Review = db.model("Review", reviewSchema);  
module.exports = Review;
```

#### 4. Express Route Implementation (10 points)

Given the below `PhoneNumber` model and JSON response format, complete the `/isrobo` endpoint that lookups a phone number to determine if the phone number is robot caller (i.e., robocall).

1. The phone number is provided in the query string of a GET request using the key `phone`.  
Ex: `http://somedomain.com/isrobo?phone=8005551212`
2. If the phone number is found in the database and has more than 100 robot caller reports
  - a. The response's `roboCaller` property will be `true`.
  - b. The response's `numReports` property will be the number of reports.
3. If the phone number is found in the database and has less than or equal to 100 robot caller reports:
  - a. The response's `roboCaller` property will be `false`.
  - b. The response's `numReports` property will be the number of reports.
4. If the phone number is not found in the database:
  - a. The response's `roboCaller` property will be `false`
  - b. The response's `numReports` property will be `-1`.
5. The endpoint should return an appropriate response code.

##### PhoneNumber Schema:

```
var db = require("../db");

var phoneNumberSchema = new db.Schema({
  number: String,
  name: String,
  robocallReports: Number,
});

var PhoneNumber = db.model("PhoneNumber", phoneNumberSchema);

module.exports = PhoneNumber;
```

##### Response (Number found, Robot caller):

```
{
  roboCaller: true,
  numReports: 1105
}
```

##### Response (Number found, Not a robot caller):

```
{
  roboCaller: false,
  numReports: -1;
}
```

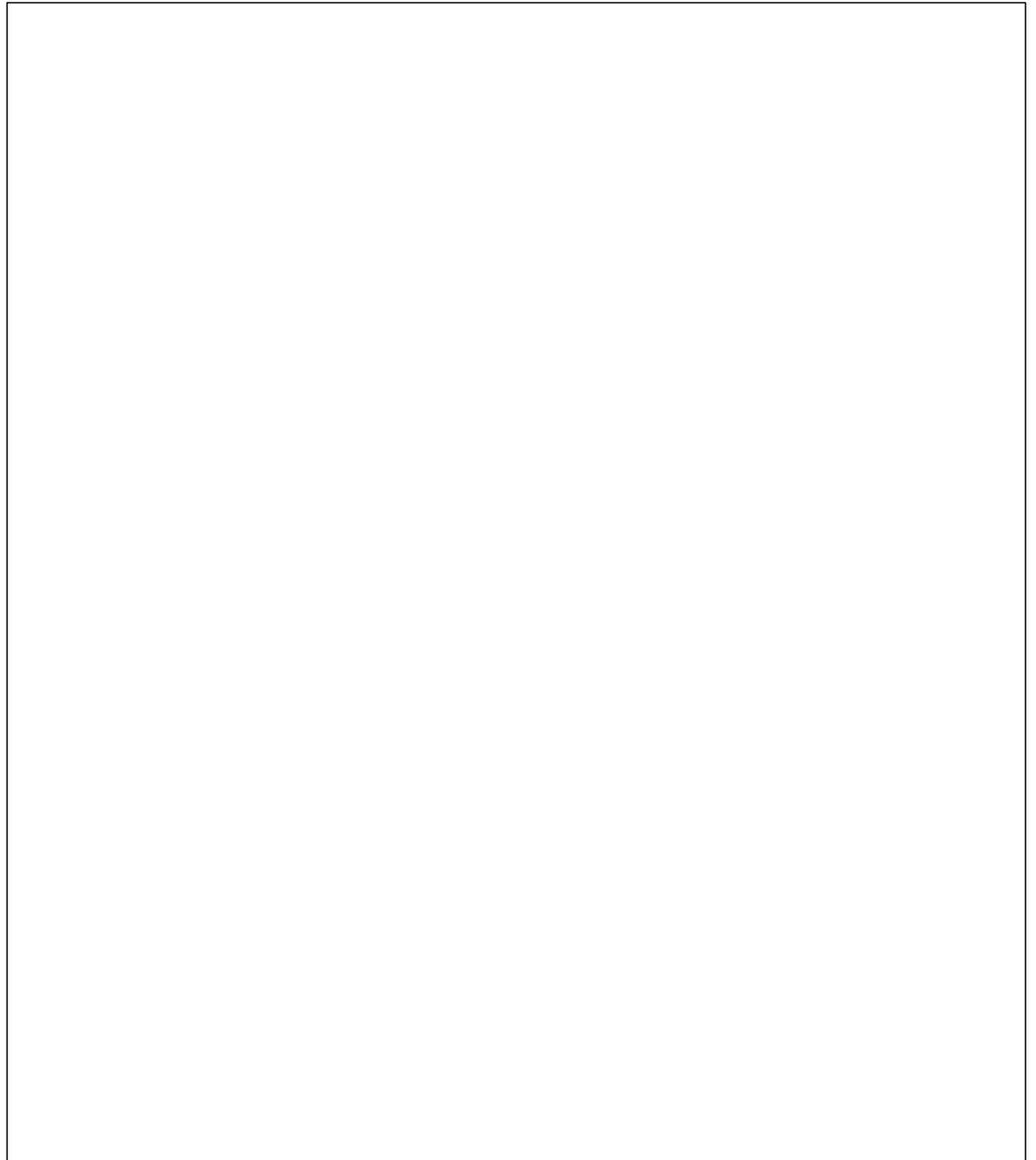
##### Response (Number not found):

```
{
  roboCaller: false,
  numReports: -1;
}
```

Write your code in the box on the next page.

```
var express = require('express');  
var router = express.Router();  
var PhoneNumber = require('../models/phonenumber');
```

```
router.get('/isrobo, function(req, res, next) {
```



```
});
```