(100 pts - 20 min)

Vanderbilt University

CS 4288: Web-Based Architectures

**Quiz Five - 22F**

**Instructions:**

* Submit your work to Brightspace under Assignments/Quiz5
* Name the file with your vunetid
* Do not hand write your answers!!!
* Answer the following questions in your own words, DO NOT JUST COPY PASTE.

1. (20 *points*) Look at the Express route handlers setup in order below. Match the appropriate handler for each requested path. Not all handlers are necessarily used. Also, provide what the URL parameter(s) passed to the handler will be:

app.get(‘/v1/user/:username', (req, res) => { … }); // Handler #1

app.post('/v1/user/:username', (req, res) => { … }); // Handler #2

app.put('/v1/:username', (req, res) => { … }); // Handler #3

app.put('/v1/user/:username', (req, res) => { … }); // Handler #4

app.get('/v1/email/:address', (req, res) => { … }); // Handler #5

app.head(‘/v1/:user/:username', (req, res) => { … }); // Handler #6

app.get(‘/v1/emails?q=10’, (req, res) => { … }); // Handler #7

app.post(‘/v2/user/:user', (req, res) => { … }); // Handler #8

app.post(‘\*’, (req, res) => { … }); // Handler #9

* 1. POST to /v1/user/heminggs \_\_\_\_\_\_\_\_\_\_\_Handler #2,param :username=heminggs
  2. PUT to /v1/user/heminggs \_\_\_\_\_\_\_\_\_\_\_Handler #4,param :username=heminggs
  3. GET to /v1/user/eagle#bar \_\_\_\_\_\_\_\_\_\_\_Handler #1,param :username=eagle
  4. DELETE to /v1/user/tumbler \_\_\_\_\_\_\_\_\_\_\_none
  5. POST to /v1/heminggs \_\_\_\_\_\_\_\_\_\_\_Handler #9
  6. POST to /v2/user/heminggs \_\_\_\_\_\_\_\_\_\_\_Handler #8,param :user=heminggs
  7. HEAD to /v1/user/heminggs Handler #6,param :user=user, :username=heminggs
  8. GET to /v1/emails \_\_\_\_\_\_\_\_\_\_\_none
  9. POST to /v1/emails \_\_\_\_\_\_\_\_\_\_\_Handler #9
  10. POST to /v1/mail/forward \_\_\_\_\_\_\_\_\_\_\_Handler #9

1. (10 points). When creating a cloud-based VM, you typically must determine:
   * SSH keypair, access port, network path, and router configuration
   * Cheese, pepperoni, veggie supreme, or stuffed crust
   * Speed, duration, frequency & connection state
   * **Location, size, operating system, access & security**
2. (10 points bonus). Where does Salvatore Sanfilippo live:
   * Modena, Emilia-Romagna, Italy
   * **Catania, Sicily, Italy**
   * Ponza, La Forna, Italy
   * Campobello di Licata, Italy
   * East San Jose, California, USA
3. (20 points). What is the primary advantage of using a service like Redis for query caching? What is the primary disadvantage?

The primary advantage is that after the data is cached, it is faster to read from the cache than it would be from the database. The primary disadvantage is that it takes time to update the cached data, so it may be old data that the user gets. It also takes longer on the first request because the program will check the cache first and find nothing.

1. (20 points) Why would a user not want 2-Factor authentication enabled for their account on a security sensitive website (banking for example)? What can you do to encourage them to adopt it? What would be the next best alternative means of increasing authentication security?

A user may not want 2-Factor authentication because it is an extra step required to login and also would lock them out if they forget either Factor. It could be encouraged by showing how much easier it is to hack an account if it only needs username and password. The next best alternative might be something like using SSO.

1. (20 points) Where does each technology get used (client, server or both), and when is it used (build-time, run-time or both):
   * MongoDB: server, run time
   * Webpack: client, build time
   * React: client, both
   * Mongoose: server, run time
   * Redis: server, run time
   * Express: server, run time
   * Styled-Components: client, both
   * jQuery: client, run time
   * Babel: client, build-time
   * Bootstrap: client, run time
   * HTTP: both, run time
   * TOTP: server, both