CSC309: A5 Report

a.
Title: KiBay
Members: Harris Fok (g5harris), Fiona Lo (g3flo), Kyla Palos (g3palosk), Minsoo Park (c4parkmj)
Description: KiBay is a centralized online community for local classifieds and advertisements that works similarly to its competitors, Kijiji and Craigslist, where users post ad listings of their products or services and buyers contact them through independent communication channels. With Kibay's recommendations system, a typical user will be able to judge the quality of the product or service provided and may email the ad's lister to purchase or inquire about it. A seller may also have a location listed in their profiles should the buyer wish to judge the distance required for pickup or delivery. This is a sharing economy as users share access to goods or services while optimizing resources through redistribution and reusability.

b. **Design:**
(i) http://i.imgur.com/UAa9KAz.png

How it Works
For those that wish to create their own listings, they must first register and log in with their email and a password. Buyers are welcome to create an account as well but are free to browse and search the listings without one as a guest.

The first user to sign up is automatically the super administrator and has the privilege of removing other users, their listings and promoting other users to be administrators. However, they cannot delete their account. The promoted administrators perform the same as the super administrator but are allowed to delete their account.

When going to their profile for the first time, users will only see their email. However, they can add additional information with the edit profile button in the center of the page. This includes an icon of themselves which is recommended to be from a small sized image. By registering an account with the site, users also gain an additional ability to comment on other listings on the site.

(ii) For simplicity, we decided to use the Mongoose design pattern for MongoDB, written in node.js to implement our server side validation and casting. The overall idea of KiBay requires two main schemas, which can be implemented thanks to Mongoose: these schemas are users and listings. Additionally, we used Sessions for authentication and management and the Jade view engine for designing our views. The database holds the users and listings while the routes access them for use in their related functions which include, creating, editing, removing users and listings as well as many other backend functions. The routes then link to the respective jade views for page transitions, form retrieval etc.

(iii) Main Files:

(database) - app.js

public:

- images
- javascripts:
  - listings.js
  - profile.js
- stylesheets:
  - index.css
  - listing-edit.css
  - login.css
  - profile.css
  - style.css

routes:

- index.js
- listings.js
- search.js
- users.js

views:

- error.jade
- index.jade
- layout.jade
- listings-edit.jade
- listings.jade
- login.jade
- profile.jade
- search.jade
- signup.jade

c. In KiBay, security vulnerability is tested using authentication and authorization. Identity verification and authorization is in place with the use of passwords to log in as well as a backend sessions manager (using Express.js sessions framework) to verify that the user has the correct credentials and login status. Confidentiality is also in place as users are only asked for the emails upon signing up and any additional private information they provide is at their discretion. As users do not pay through KiBay and the messaging system is through their own private email, all money transactions and communications may done on the users' sides although they are responsible for their own repudiation.

d. We use basic cache control in express.js for site images. This allows for temporary storage of our image paths so that they do not need to be continuously retrieved from the server at every page load. From this, we reduce bandwidth, server load and page waiting times.

e. **(video link)**

f. A link to our code repository can be found here: https://github.com/Kitsunemimi/Echelon

Note: Kyla & Fiona worked together for most tasks. See github issues and comments for additional task breakdowns.

Features that were started but not fully implemented due to time constraints:
These features can be found in respective branches of the git repository linked above.

- Facebook authentication. Updating the users schema to handle facebook logins.
- A page for email contact between buyers & sellers (route available but no functionality). Ideally, the page will allow users to type out a message and have the site send the content as an email to the intended recipient’s mailbox.
- Additional optimization techniques. Using locust.io to graph performance figures.

Features that were planned but not implemented due to time constraints:

- password hashing.
- sorting listings by category.