SWAG DEPOT Report

Git: https://github.com/Rbandon64sp/CSC309-A4

If you are not able to access the repo please contact me at
Brandon.sherman100@gmail.com or Brandon.sherman@mail.utoronto.com

Members:

Brandon Sherman
CDF: g5sherm
Git username: Rbandon64sp

Vibhavi Peiris
CDF: c3peiris
Git username: vibe95

Duy Le
CDF: c3leduyt
Git username: dz14

Note: We did not have a four-person team for phase 2 because our fourth member dropped the course.

Our initial plan in the project was to design a website where people would be able to view and rate other individual's photos and styles. We were hoping to create an environment where if you were able to show your lifestyle to other people, this would inspire other people with fashion choices they could make. Due to the workload and the fact that our fourth member never notified us until the 2nd week from the due date we were only able to do the barebones of the project. If we had more time to work on this there are certain decisions we would have liked to change.

We designed the front end using jade's template engine. We used bootstrap, and an external party's software called fblightbox. This allowed us to implement an animation for photos on our site. As required we used NodeJs for the backend and fulfilled the other software requirements.

Backend Design:

We implemented a mvc with jade where different views populate layout.jade with their information so each view can be rendered onto the front end. We designed all of our APIs inside of the folder routes and we used express for the functions. We organized each file into their own sections with bin containing the server code, public containing our css/stylesheets, javascript, data holding our database. All required packages are obtained through ‘npm install’. They are listed in our package.json and app.js links all of our software together.

The features from our initial design that we were successfully able to incorporate were:

- User registration
- Log in
- Log out
- Individual profiles
    - Viewing this information
    - Editing information
    - Uploading photos through profile
    - Viewing the user’s personal photos
    - Profile picture
- Viewing other individual profiles.
- Searching for users through a search bar (using username)
- Our photo stream
    - Streams photos to the person viewing it from the database.
    - Commenting on user photos
    - Rating

To ensure security we created a session variable that would store the user from the database onto the cache. This allows us to ensure user profiles could not be changed by another user unless that user is able to login to the account.

The features that we were unable to implement because we were pressed for time and the amount of time it took to figure out how to successfully upload and render photos from our database took quite a bit of time.

Issues:

- Admin user
    - We were able to implement some of the backend, but did not have time to create a front end. It is there under /routes/admins.js
- Social Network
    - We were planning to make some form of chat-based system between users. Users can only chat through the comment section of photos.
    - We also were thinking of implementing some box on the user page for messages.
- Photo Stream
    - We were planning to limit the amount of photos that were being rendered unless the user wanted more but currently we ran out of time. This is the same with user photos as well. Selecting the user

profile picture we have the option, but it is not currently implemented.

The workflow of how to use the site is detailed on the readme.pdf.

Security:

The basic security that we implemented was using module crypto cipher and decipher on passwords. This ensured that the user credentials was secure, since even if people were able to access the database they would require the decryption key. We wanted to also want to implement sanitize, but that was not on priorities, since we were pressed for time.

As I stated earlier in the report user account information cannot be changed unless the user is logged in.