

Project 4 — Microblogging

For Project 4, you will be implementing a basic Twitter-like service that lets users share short text and pictures, and organizes the results into a news feed.

This project is due on **April 12, 2016**.

Note:

You may know that I was originally planning for this to be a mobile-friendly Snapchat application. Unfortunately, accessing mobile phone cameras is significantly more difficult than I had realized in early planning, so I must alter the plan.

Requirements

Create a web app that supports the following:

- Users can register accounts and log in. Each user needs to have a user name and password, and can have a Location (short textual description of their location), a Bio (a short blurb about where they are from), and an avatar picture. Do not allow usernames to contain spaces, /, or other punctuation characters.

Hint:

A regular expression can be useful for doing this validation.

- Users can edit their profile to change the bio, location, or upload a new avatar picture.
- Users can view other users' profiles and *follow* them. This should be implemented through *profile pages* available at the URL `/u/<user>`, where `<user>` is the user's username. If the viewer is a logged-in user, show a 'Follow' button on the profile page, and record that the logged-in user now follows the displayed user when the button is clicked. If the logged-in user already follows the displayed user, the button should be 'Unfollow' and it should remove the follow relationship.

This logic should be implemented with JavaScript responding to the button click, posting the appropriate request, and changing the button's state and text when the request is successful.

Hint:

Store user-follows-user relationships as a separate class in your model, `Follow`, that has the follower ID and followee ID.

- User profile pages should show the user's 10 most recent posts, in reverse order of

date.

- The home page (/) should display one of the following:
 - If the user is not logged in, show the 20 most recent posts across all users and allow the user to register an account.
 - If the user is logged in, show the 20 most recent posts from the users they follow.
- Allow users to create new posts. Posts should be limited to 256 characters, in addition to an optional URL (this is a deviation from the Twitter model). Implement posting via JavaScript as well; when the post has succeeded, add it to the top of the timeline.
- When displaying posts, either in the main news feed or in a user's profile, display the following:
 - Posting user's name
 - Posting user's avatar picture (restricted to be small, 64×64px)
 - Date and time the message was posted
 - Post text
 - URL (if provided), as a clickable hyperlink

Posts should always be displayed newest-first.

Implementing these requirements correctly will earn a solid B. To earn an A, add:

- Messages can have an accompanying *photo*. To implement this, you may need to use a traditional HTML form POST instead of JavaScript.
- If the post has a photo, display a (size-restricted) version of it with the message.

Deploying on the Server

For this project, we're going to add server-based deployment. Deploy your application on your Azure virtual machine, accessible via both HTTP and HTTPS, and submit the following to TRACS:

- The code (from `manage.py` package) as an attachment.
- The URL to your application (e.g. `https://txst-g99-p4.southcentralus.cloudapp.com`) in the TRACS text.

Additional Things to Explore

Some additional things you can consider adding if you have time & inclination:

- JavaScript-based assistance for resizing photos and avatar pictures.
- Location where the message is posted, using the browser's geolocation API.

redistribution is prohibited without prior consent from the author.