

# HW2: Twitter v192

Due March 28th at 11:59 PM

## Introduction

The beauty of web development is its ability to create software that anybody with an internet connection can use. In this assignment, you will implement a clone of my favourite social media - Twitter! An implementation of the website is available [here](#) and you are expected to recreate all the functionality from the site. This might seem a little daunting - but I assure you that everything you need to know has been covered in lecture.

In particular, you must be able to sign in and create tweets that support hashtagging (i.e. the tweet "I love #programming in #cis192" contains two hashtags) such that accessing the page for a given hashtag will return the tweets that contain the hashtag. In addition, you must be able to delete tweets (if you are signed in as the author). You also need to create profile pages for each user that display their username and the tweets they wrote (all tweets must be displayed in order of recency). For this assignment, all tweets will be public (in essence, all users follow each other by default).

This assignment is meant to be developed from scratch, including importing packages from the [pip](#) registry. Feel free to consult the Django boilerplate code provided in lecture. Collaboration with one other student is permitted but all students must submit their own assignment.

## Milestone 1: Templates

The recommended approach to developing this application is to start by writing the HTML templates that the application requires. At the very minimum the application requires the following pages:

1. Log In and Sign Up page(s)
2. Splash page (explanation of product)
3. Home page (feed of tweets and hashtags)

- 4. Profile page (display of each users tweets)
- 5. Hashtag page (display of the tweets that correspond to a hashtag)

You should also write the URLs and views to simply render the templates too, in order to both debug and set up some boilerplate for you to implement in Milestone 3. Be sure to use a CSS framework to keep your web pages production-ready. I like to use Bootstrap or Bulma, but anything that doesn't use Times New Roman is fair game.

## Milestone 2: Accounts System

Once you have a working accounts system, the entire flow of development will be completely different. Django's authentication system lets us implement this very easily (refer to Week 7's lecture notes for more information). It is not necessary to extend the user model (e.g. bio, profile picture etc.), but extra credit will be given where due!

## Milestone 3: Model Definitions

The only two models that you need to implement are a **Tweet** model and a **Hashtag** model. Be sure to make use of SQL relationships (namely the **foreign\_key** and **ManyToMany** relationships). Tweets must contain a timestamp and their associated author. In addition, tweets must supporting the "liking" feature such that each user can like and unlike tweets.

Be sure to create a user using **createsuperuser** and use the admin page to test out your models before moving onto implementing your views.

**HALF WAY POINT (<https://www.youtube.com/watch?v=EBZp9IZFdTU>)**

## Milestone 3: Views

It's time to put everything together and make your application functional! Create the associated views that correspond to the templates and forms that you wrote in your HTML.

## Submission and Grading

This assignment will be graded manually and will be evaluated on correctness and overall project design (including style). Create a file named **README.md** in the root directory of your

project. This README file should document your routes, any design considerations you made, any collaborators, and any extra credit. The README file should also include information on how to run the server. Be sure to `pip freeze` into `requirements.txt` so the staff know what third party packages are used.

Finally, zip the project and submit through the Canvas site.

## Extra Credit

We will be giving extra credit for **any extra functionality** based on effort required to implement. Some ideas include:

1. A follower system (with follower list) (15%)
2. Replying to tweets (10%)
3. Deploying to a cloud service (10%)
4. Extending the user model (5%)
5. Blocking/Unblocking users (5%)
6. Error handling (5%)

## Best of 192 Award

The CIS 192 staff will be selecting one project to win our **Best of 192 Award**! The winning project will be hosted (for free!) on the 192 server and will be listed on the course website. In addition, you will be given a 15% bonus on your assignment.