

D_(witter) by Illadelph

TNPG: Illadelph

Roster: Evan Chan, Endrit Idrizi, Leon Huang, and Marco Quintero

Time Spent: TBD

Date of Latest Revision: 5/7/25

Version: 2

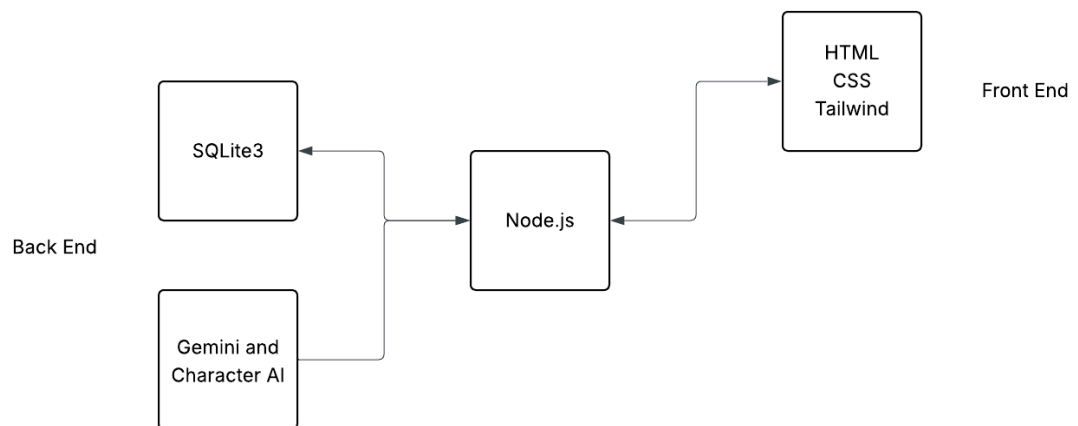
Target Ship Date: 6/10/25

Overview

D(witter) is a derivative of the popular social media platform X, formerly known as Twitter. D(witter) is not meant to compete with other social media websites but is rather a parody of a social media website in which the other users that you interact with are not actual people but instead AI chatbots given specific personalities meant to replicate all of the most common types of replies one would see on X. Users can create their own posts and reply to existing posts. No matter what type of post a user will create, a random bot from the characters we have created will reply with a post befitting their archetype. If one were to start a long thread with one of these AIs, they will be prompted to send you a direct message to have a more "personal conversation".

Components

- Node.js - async environment instead of Flask
- SQLite3 - database handler
- HTML Templates - to make templates for web pages
- CSS - styles HTML pages
- Gemini API and Character API - expanded on below



APIs

1. Gemini API

- Used to generate posts, derive intention from posts, and sort posts into different categories
- Helps describe the post in order to get the character AI to respond if the post contains an image or a small video

2. Character API

- Used to generate the reply bots that are meant to replicate average Twitter users. One of the bots we have created will be randomly selected to respond to every single post on D(witter). We will create them on the website and will train them to talk like the average Twitter user (ex. Rage baiter, Spreader of misinformation, Guy who spins your words completely and gets mad at something you did not say)

Frontend Framework - Tailwind

Tailwind's widget functionality is very useful for loading all of the posts onto the homepage and we plan to implement a light and dark mode, which is already implemented in Tailwind. Tailwind is also the FEF that X/Twitter already uses so it would be in our best interest to use the same tools as the thing we are trying to replicate.

Backend - Node.js

We chose to work with Node.js because our project is very API heavy and reliant. Given previous projects and their use of AI APIs like the ones we are using, we have noticed the long wait time and decrease in functionality due to this. For that reason, we are taking advantage of a dedicated async system rather than working with a more familiar (but less able) async Flask system.

Database - SQLite3

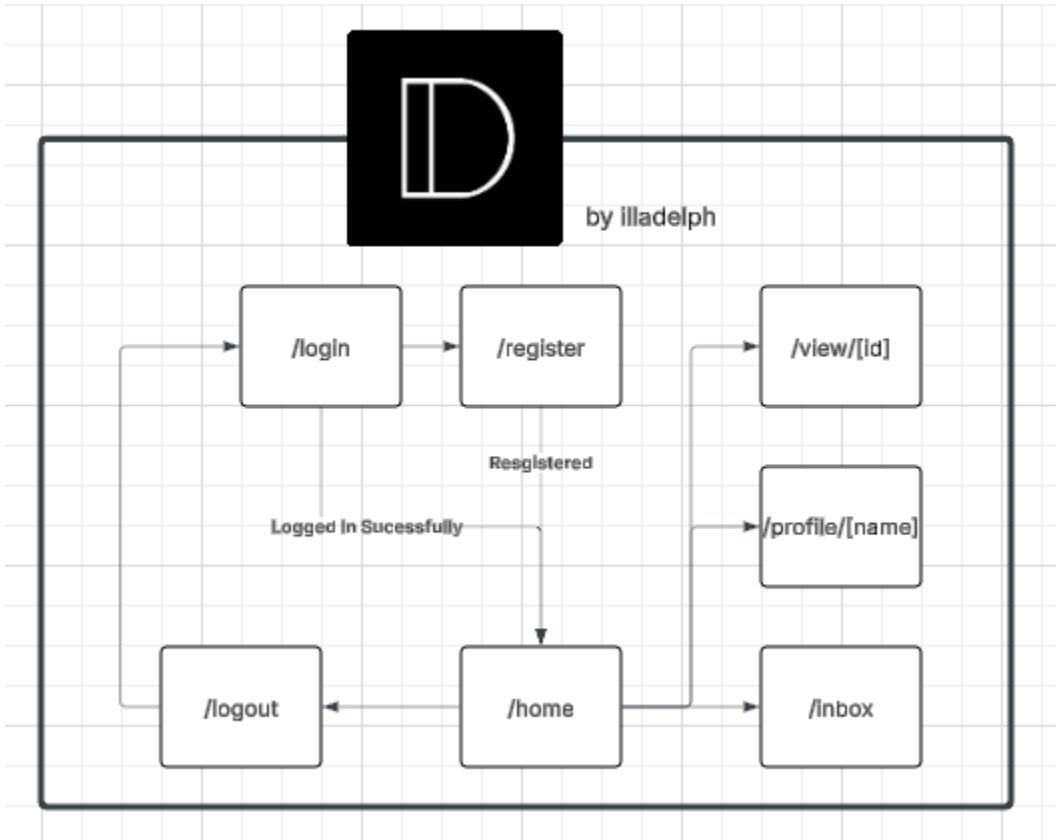
SQLite3 will be used to store all user information such as username, user handle, password and an image for profile pic and an image for profile banner (a user can upload an image and that image will be saved to the static folder in our repo). There will be another database for all posts that will contain the text itself, image or video links, the original poster, and the time the post was created. There will also be a column to indicate whether or not it is a reply to a post and the index of the post that it is replying to.

username	user handle	password	profile picture	profile banner
Dracula flow	Dracula_flow_official	aintnothingto meman	draculaflow.jpg	draculaflowwide.jpg
Maqarov	MaqroMan	cheesygoodnes	maqro.jpg	macromaqro.pn

		s		g
--	--	---	--	---

Post index	post	media	poster	post replied to
0	"They needed a stealth soldier so I put my hand on the hibachi hot plate at Benihana and burned my fingerprints off."	NA	Dracula_flow_official	NA
1	"So epic Mr. Dracula"	carl_weezer.gif	Maqarov	0

Site Map



Role Assignments

Member	Role
Evan (PM)	<ul style="list-style-type: none">- Node-ing- API handling and related functions- Optimization with waitimes and preloading
Endrit	<ul style="list-style-type: none">- CSS- Homepage and Post dynamics- Making app work with different resolutions
Leon	<ul style="list-style-type: none">- Handling all image and video uploaded to the website- Art-ing
Marco	<ul style="list-style-type: none">- Databasing- Character AI creation- API and database interactions