

Progress Report

- Increment 1 -

Group #3

1) Team Members

David Markel, dm20gg, david-markel

Victor Marques, vfm20e, VFM20E

Dylan Gire, dog18, dog2943

Tommy Chong, tc18cj, toch24

Abraham Beltran, ab19bb, abrahambeltran

2) Project Title and Description

DataTweet, a web app for sharing visualized data similar to Twitter where users can post any data they may want to share like research results with visualization options. It is using React for the front end Django for the backend, and sqlite for our database.

3) Accomplishments and overall project status during this increment

We built the basic frontend components of the web application, this includes the homepage, the about page, and the profile page. We set up the database in the backend with the Users and Posts table. Alongside the database the Django app DataTweetApi was established. This app contains the models (User and Posts) serializers, urls and views in order to connect the information from the database. We then were able to connect the database to the frontend and created one crucial feature which is posting. Users that are logged in will be able to post on the home page, once a user clicks on “post”, the data is stored in the database. We also worked partially on the login/registration system. Overall status of the application: Basic frontend UI implemented, posting is implemented, a database is set up, and login/registration system is being worked on.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

Some challenges that we encountered was the setting up of the database. Since Django connects to the database via migrations, it wasn't the traditional way we used to set up the database for our applications. Thus, having to learn this was a challenge as we were running into a lot of errors. In our original proposal, we said we wanted to use Firebase for our database. We have changed that to SQLite because we didn't have much experience with Firebase and it was becoming a nuisance to implement. Since SQLite is already integrated with Django, we decided to go for SQLite as our database.

Some things that went wrong during this increment is fetching the feed with the posts data. Tommy was implementing this part and unfortunately, he wasn't able to make it work fully as he was running into a lot of errors when parsing the data from the backend and formatting it into the feed. Thus, the data is being fetched in the homepage from the backend. However, it is incomplete.

Other challenges were seen with authentication due to the strange behaviors that occasionally happen between Django and React. In this instance it was more of a split decision to cut out complexity with auth0 and settle for something else that would work for our current use case. This may or may not be our final solution to authentication but it depends on what is needed in the future for logging user data.

5) Team Member Contribution for this increment

- Tommy Chong:
 - A. Added to part 3 on the contributions I made as an accomplishment for this increment and 4 on the challenges I encountered on the contributions mentioned. Listed what I am doing for the next increment in part 6.
 - B. Mentioned and described the functionalities in part 2 from 1-6. Mentioned and described the functionalities in part 3 from 1-3. Added the React section to part 6 and added the React dependencies for part 7.
 - C. Added more descriptions to the languages to part 1, added the frontend section to part 2 and described our testing in part 5. Added the functional testings in part 3 and added the Perf library description for part 4.
 - D. Designed the home page with CSS to make it look aesthetically pleasing. Helped out in the set up of database by fixing a bug with the paths and adding and modifying some columns to the Posts model. Implemented the functionality of writing a post in the homepage and posting it to the database. Worked in displaying data from the Posts table into the Homepage feed.
 - E. Recorded the video for the demo.
- Dylan Gire:
 - A) Added to 5 and 6 on specific contributions I've worked on.
 - B) Added to 1 and 2 regarding the search contribution as well as some non-functional requirements that would work best with the program going forward.
 - C) N/A
 - D) Implemented the beginning of a search component that is designed to search the postings with a specific keyword. Need to implement an automatic pull from the database to search and also make it more aesthetically pleasing.
 - E) N/A
- Victor Marques:
 - A. added to 3 on Backend development, 4 with issues on firebase implementation
 - B. Created the two diagrams (case and sequence)
 - C. (1) added the languages and initial descriptions later fleshed out by others (2) added backend description of DataTweet, DataTweetApi, and Encryption.
 - D. Backend: Added the database initializer to allow setup of main users, set up the DataTweetApi with Django framework (Serializers, urls, views, models). and added an encryption/decryption file from PyCrypto for security.
 - E. N/A
- Abraham Beltran:
 - A) added to part 2, part 4 (talked about issues with authentication), and part 6
 - B) added to part 2
 - C) N/A

- D) Implemented auth0 functionality for logging in and logging out users as well as the profile view of the users, also added some backend functionality for possible future use for logging and registering users. Auth0 may be a temporary fix depending on what is needed as the project progresses, this area will most likely be revisited.
- E) no
- David Markel
 - A) Progress report: just my own contributions
 - B) Added to parts 2, 3, 6, and 7
 - C) Added on parts to 1, 2, 3, and 5
 - D) Source code organized file structured into components and styles. Created basic components and pages. Created the nav bar component. Set up system for navigating between parts without rerendering every component.
 - E) Did not work on video

6) Plans for the next increment

- Tommy: fixing and implementing the fetching of the posts data to the feed (homepage) and start working on the data visualization part of posting.
- Dylan Gire: Need to implement an automatic pull from the database to search and also make it more aesthetically pleasing. Once that is finished will continue to help debug and work on other frontend materials.
- Abraham Beltran: Will work on website themes/aesthetics, authentication, and managing user data storage alongside Victor Marques. May possibly take another look at the authentication to eliminate the reliance on auth0 in future increments.
- Victor Marques: Working on fully fleshing out the datastorage, implementing roles and definitions for them, attaching the ability to communicate with other social media platforms (share feature), implementing encryption and decryption where needed.
- David Markel: Finish up front end pages so that they all have unifying theme. Additionally, make sure pages are fleshed out in detail so that they are more than just bare bones.

7) Link to video

https://drive.google.com/file/d/1WMidfKOb30kWqrUI_w2b9HYGpYbEhViQ/view