

StatusUpdate

I implemented my application using Azure Web App, Azure Table Storage and Azure Function Apps. I used Azure Web App to deploy the application on the cloud. Azure Table Storage was utilised to store and query tweets, users and follow relations. I created three tables in total. The first table is called Users and is used to store usernames as the PartitionKey and passwords as the RowKey. Another table called Tweets stores usernames as the PartitionKey, the time of tweets in milliseconds as the RowKey and the tweets themselves as a property called tweet. The last table is called Follow and stores users and the other users that they follow, i.e. the follow relationship, as PartitionKey and RowKey respectively.

I created several different functions in the Function App to perform various tasks. One function is used to check if a user logging in has already signed up and to check if the password entered is correct. Another one is used to sign up a new user by storing the inputted username and password in the Users table if it doesn't exist. If another user with the same username exists, the user is prompted to sign up with a different username. Another function stores new follow relations in the Follow table. A function each for getting the followers and the following of the user logged in from the Follow table. One function is used to get the tweets posted by the logged-in user. Another function gets the tweets of the users followed by the logged-in user, i.e. the timeline of the logged-in user. A function for searching for tweets containing a specific keyword. A similar one for searching for users using their specific usernames. And finally, a function to store tweets of users in the Tweets table.

The application includes a variety of features. On launching the application, the user sees a Login page that has two input boxes for username and password. It enables new users to sign up for the application and existing users to login. Input validation for the username and password has been implemented to ensure strong usernames and passwords are used. Once logged in successfully, the user is redirected to their Home page. At the top left of the Home page, is a text box through which users can send out tweets of up to 140 characters. Below this section is the Timeline, where tweets of other users that the logged-in user is following are displayed. Most recent tweets are displayed at the top. The Timeline can be updated by clicking the Refresh button. Search is the optional component that I chose to implement, and it is on the top right of the application. Users can search for both tweets and other users by simply typing in characters in the search box and then hitting the enter button on the keyboard. The choice to search for tweets or users can be made by selecting the relevant tab. Users can follow other users by searching for their specific username. Below the search section, is the user bio. It displays the logged in user's current profile picture, their username, the number of their followers, the number of people they are following and a Log Out button. The Log Out button redirects the user to the Login page. Below this is the user profile section. It contains three tabs. The first tab called Tweets displays the tweets that the logged-in user has posted. The Followers tab displays the users that follow the logged-in user. The Following tab displays the users that the logged-in user is following. The profile pictures that can be seen alongside tweets and users are randomly selected from a list of image sources. They are updated every time the page is refreshed and have been added to give some colour and improve the look of the application. The tweets, in addition to the profile picture, username and text, also display the time of the tweet. If the tweet was posted on the same

day, the time of the tweet is displayed; otherwise, the date is displayed. Whenever a new user is followed, the number representing the Following, the Following tab and the Timeline is all updated. When a new tweet is posted by the user logged-in, the Tweets tab is updated.

I faced issues with using Microsoft Azure. Microsoft Azure would often not update the JavaScript and CSS for the static website for long periods, and I had to keep manually refreshing it to check if changes I made have been reflected on the application. It also took me some time to familiarise myself with how Azure Table Storage worked and how functions in the Function App could be used to query and retrieve data from the table. Another issue was sending data from the functions and tables on Azure to the frontend JavaScript. I eventually solved this by using the `JSON.stringify()` function on the context response body. Implementing the timeline of the user was also a challenging task which I solved using nested loops and the filter function. On the frontend, implementing tabs in the HTML was a bit tricky, and I had to use the help of online resources to implement them accurately. Getting the username of the logged-in user on to the home page was another issue I faced. I solved this by storing the username in a Storage object using `sessionStorage`.

I spent somewhere between 50-55 hours on this coursework. I learnt HTML and CSS and how they can be used to implement unique designs and features on the frontend of the application and how JavaScript can be used to add functionality to the frontend. Learning how to use Microsoft Azure and its various features such as Storage Accounts, Function Apps and Azure Table Storage was also interesting. Overall, it was a challenging coursework, and it allowed me to experience the time and effort that is put into making a functioning cloud-based application.