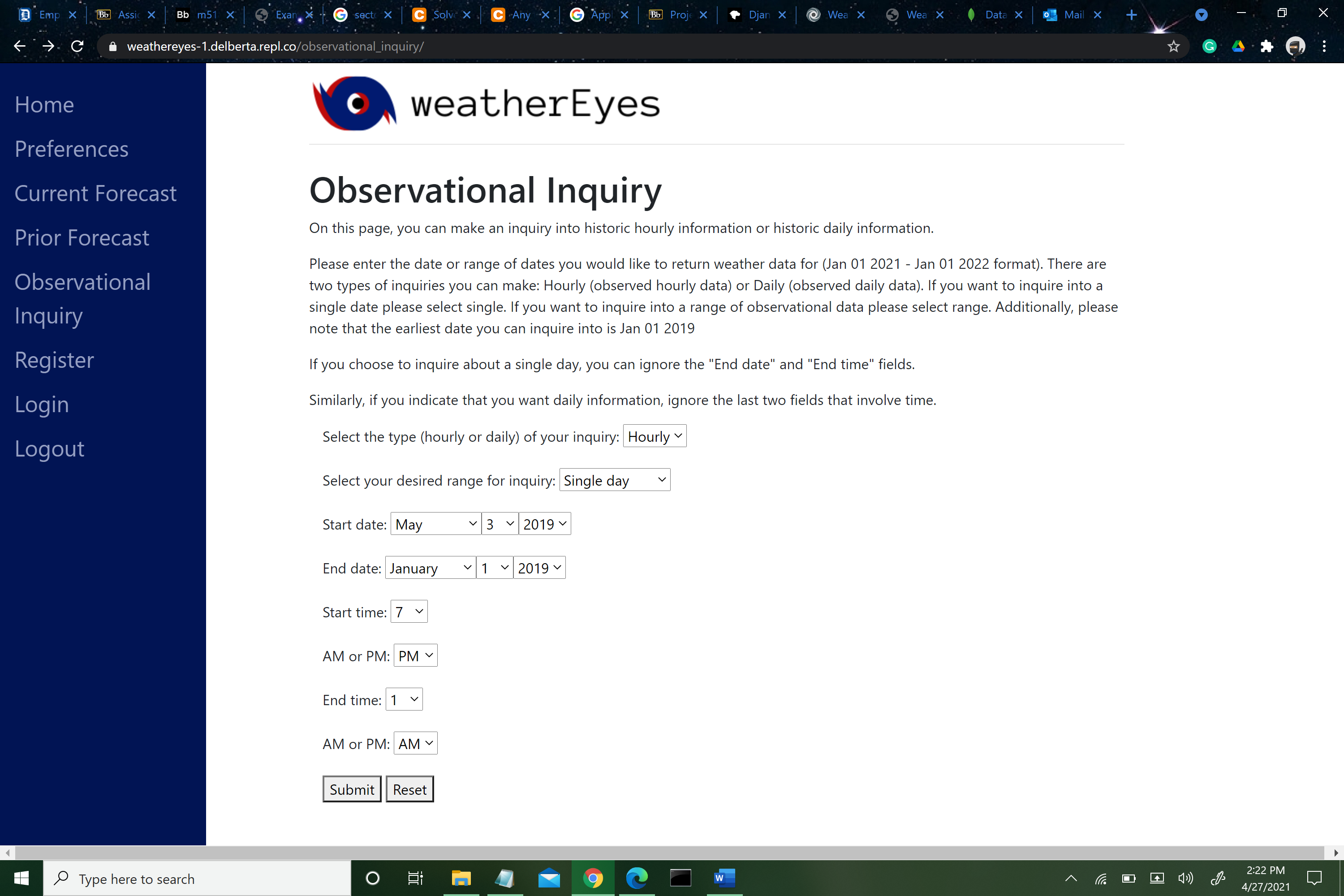
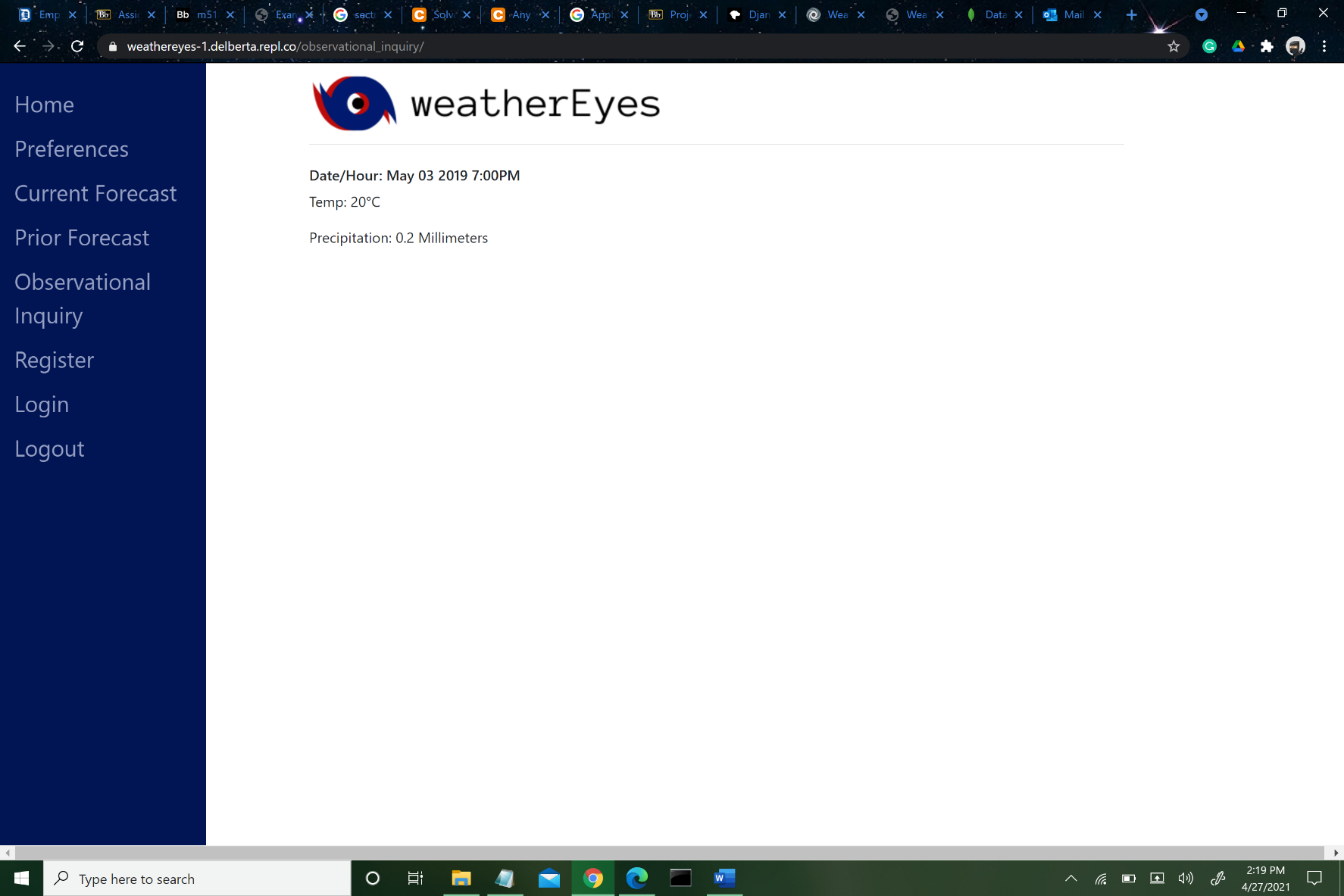
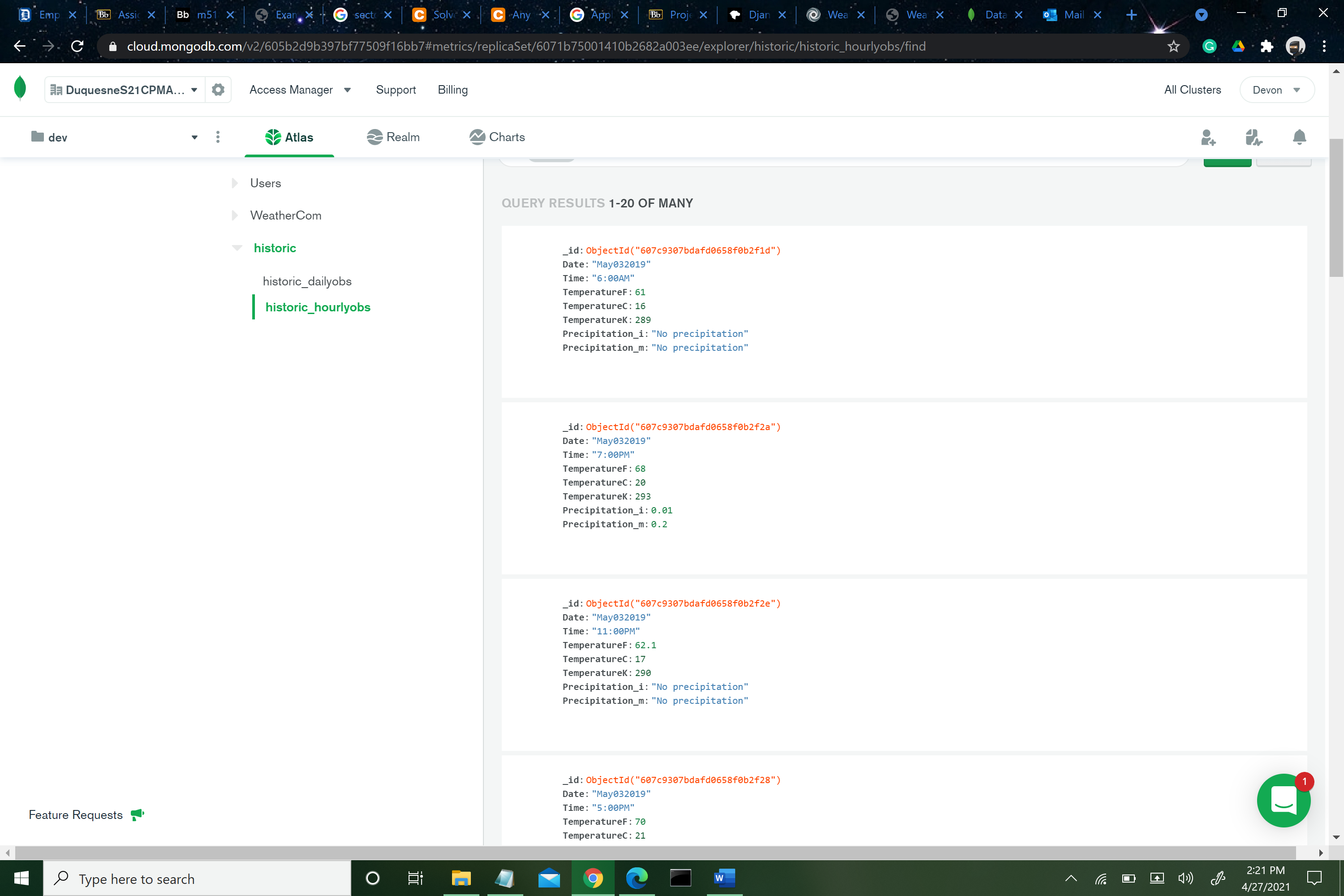
Testing in Django is not an easy task, especially on a time crunch. There is testing functionality built into Django, but I (Devon) simply have not been able to get it to work that functionality to work for our project. Similarly, Django does not seem to want to play well with pytest and throws an error every time I try to do some simple assert functions. As a result of this, I am defaulting the plan C, which is visual testing. I will manually ensure that the functionality of the weather inquiry pages are working and outputting the correct data from our database.

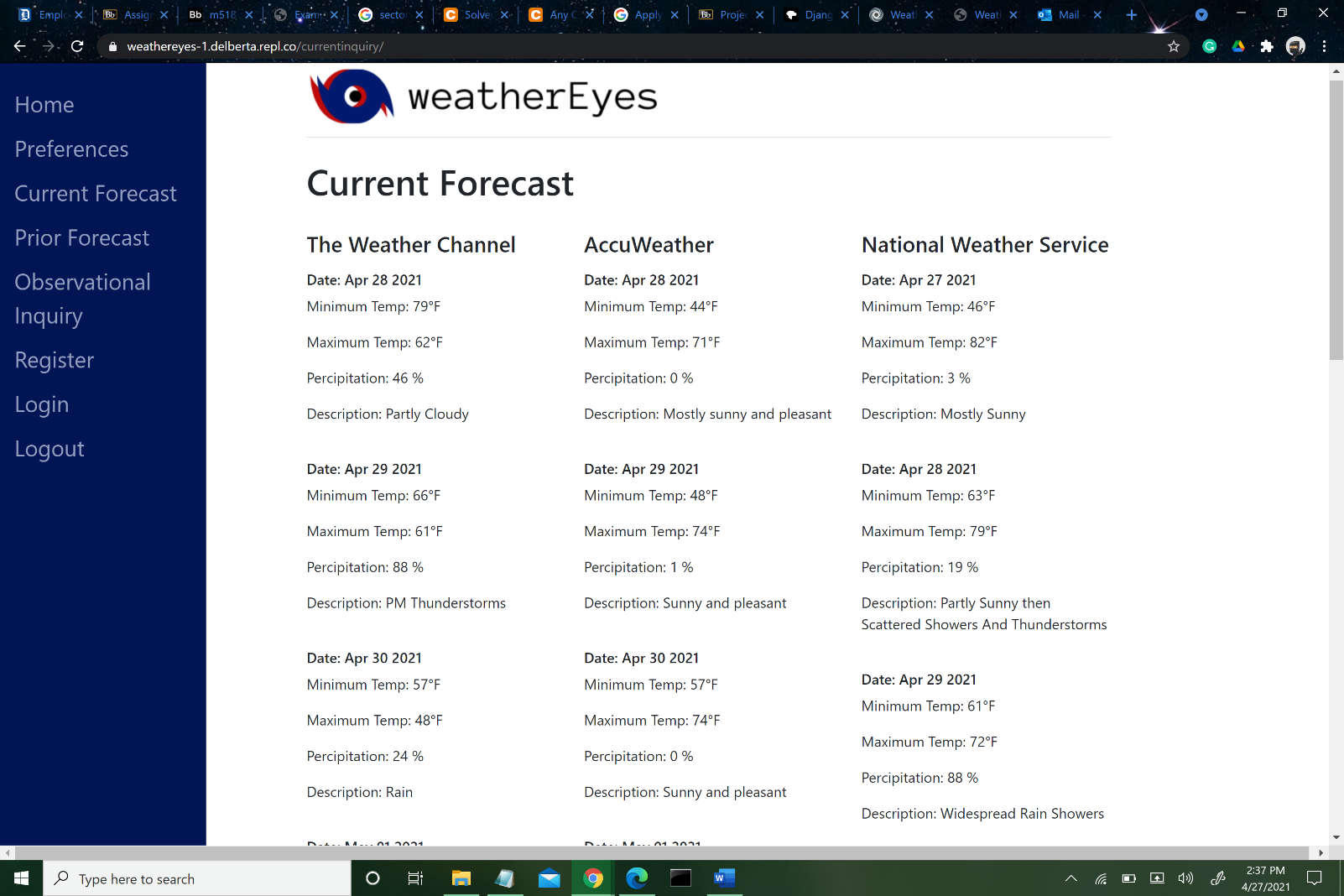
First, check the prior observations page. On the weatherEyes site, navigate to the page labelled “Observational Inquiry” and choose the options as the screenshot below.

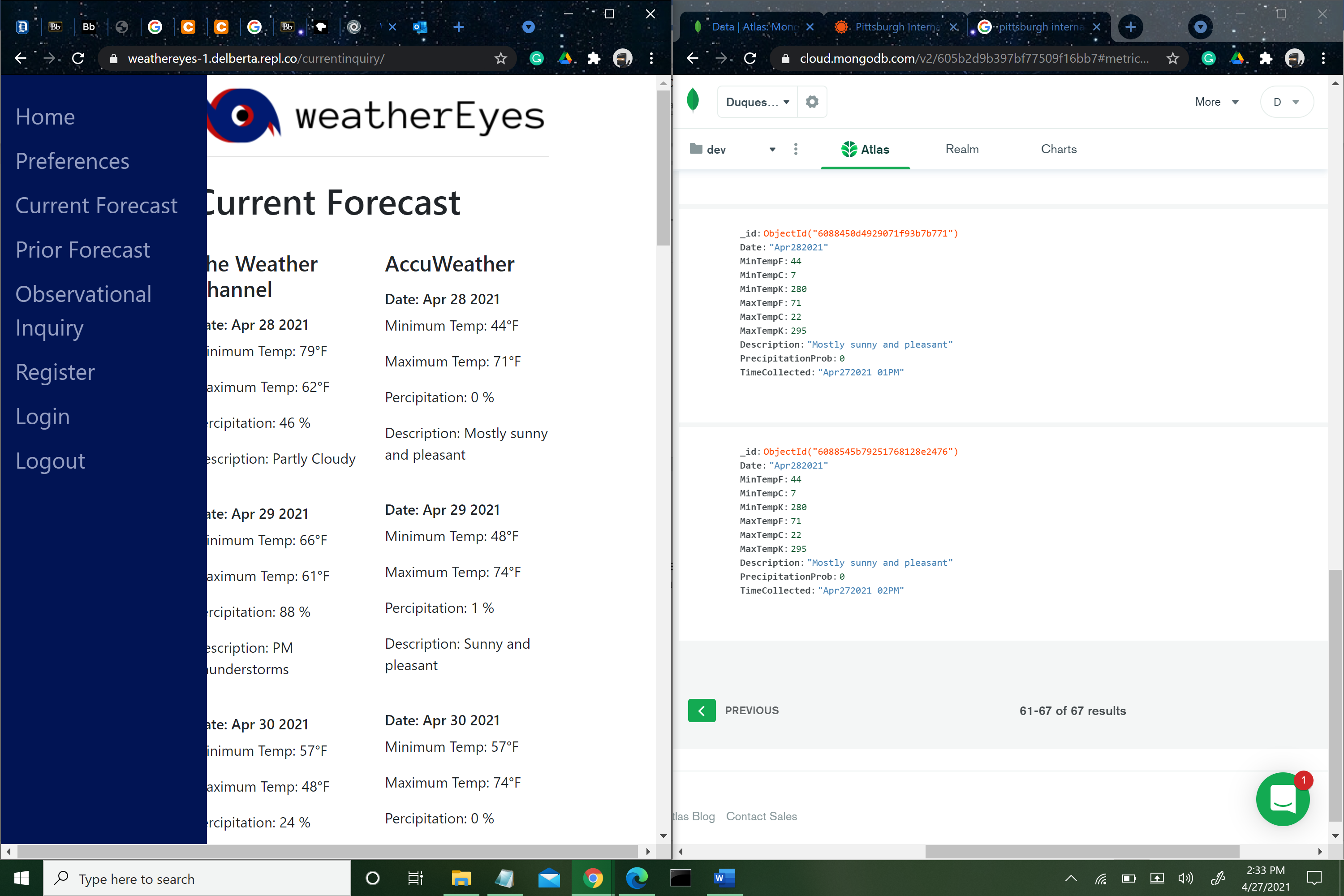


Once the appropriate options have been selected, hit “Submit” button and get output as in the righthand figure below. Visually verify that this matches the relevant Mongo database entry (which it does!)

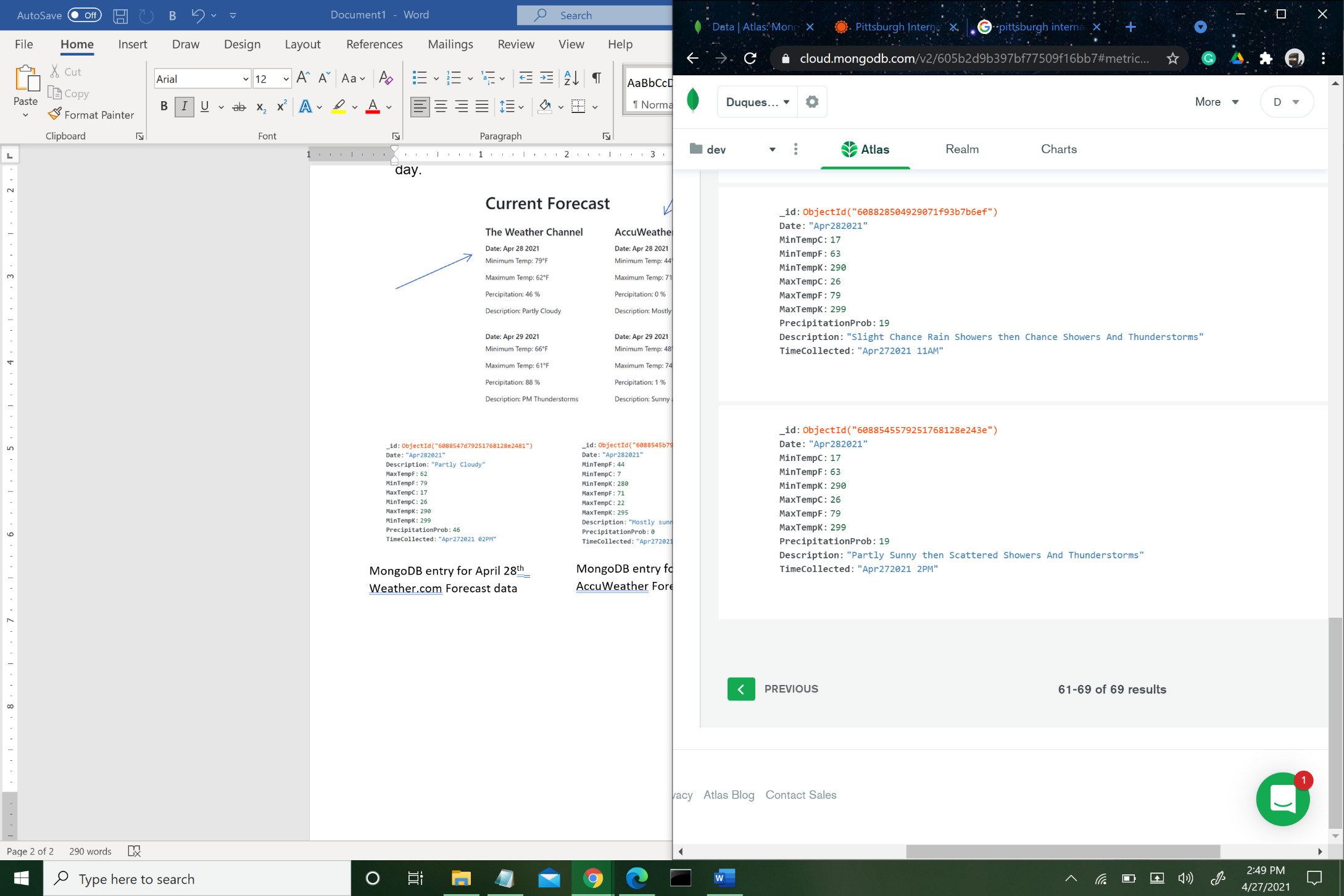


Next, check the current forecast page. Note that output on this page will be dependent upon when one visits the page and requests information. If you visit this page right at the start of a new hour, the page might not display any information for a few minutes. If this happens, try again after waiting for a little while. Navigate to the Current Forecast page and request daily forecasts. This data was checked at 2:36PM on April 27, 2021 and verified that output on the website matches the relevant databases. Note that National Weather Service daily forecasts start with the current day, whereas Accuweather and Weather.com start their current forecast with the *next* day.





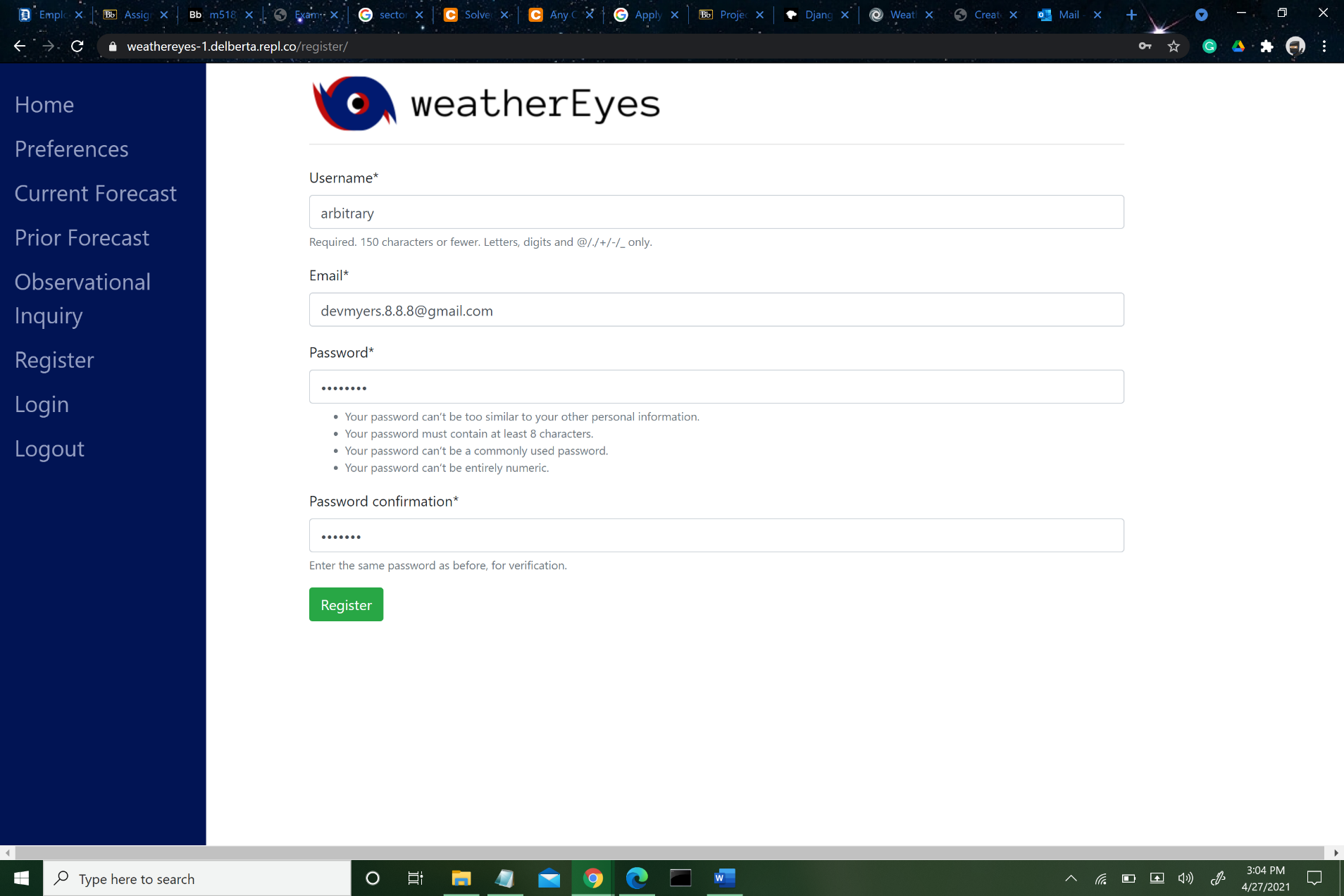
MongoDB entry for April 28th AccuWeather Forecast data

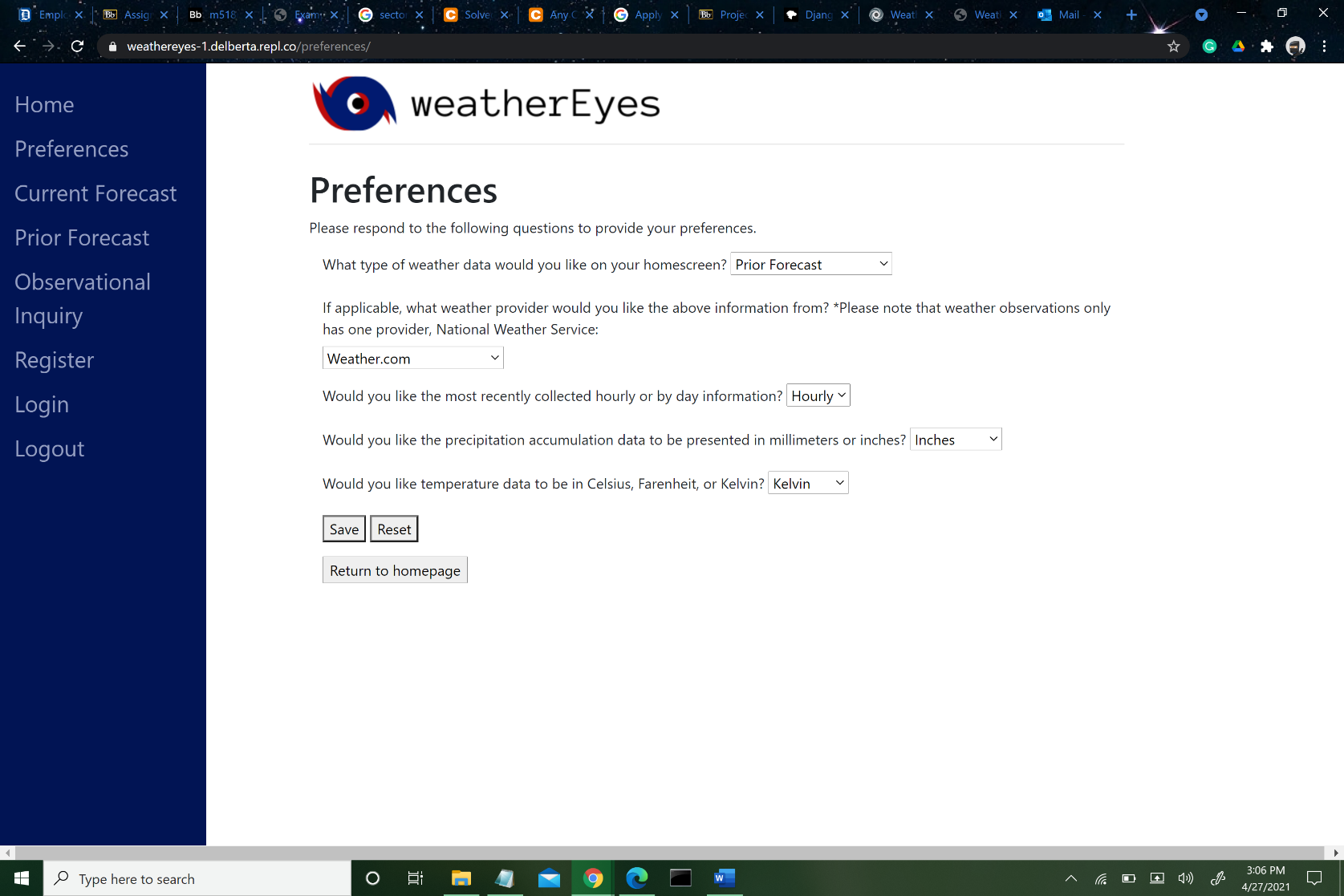


MongoDB entry for April 28th National Weather Service Forecast data

MongoDB entry for April 28th Weather.com Forecast data

Next, check that when a user is created and their preferences are set, this is reflected in the database. Again, this is just a visual test to ensure that data is flowing properly from the weatherEyes site and MongoDB. First, create a user with an arbitrary name (see left hand figure below) and then set this user’s preferences as in the figure on the right





Now, verify that the preferences were stored properly in MongoDB, which they were!

