**All the News That's Fit to Scrape**

**Overview**

In this assignment, you'll create a web app that lets users view and leave comments on the latest news. But you're not going to actually write any articles; instead, you'll flex your Mongoose and Cheerio muscles to scrape news from another site.

**Before You Begin**

1. ~~Create a GitHub repo for this assignment and clone it to your computer. Any name will do -- just make sure it's related to this project in some fashion.~~
2. ~~Run npm init. When that's finished, install and save these npm packages:~~
3. ~~express~~
4. ~~express-handlebars~~
5. ~~mongoose~~
6. ~~body-parser~~
7. ~~cheerio~~
8. ~~request~~
9. **NOTE**: If you want to earn complete credit for your work, you must use all six of these packages in your assignment.
10. ~~In order to deploy your project to Heroku, you must set up an mLab provision. mLab is remote MongoDB database that Heroku supports natively. Follow these steps to get it running:~~
11. ~~Create a Heroku app in your project directory.~~
12. ~~Run this command in your Terminal/Bash window:~~
    * ~~heroku addons:create mongolab~~
    * ~~This command will add the free mLab provision to your project~~.
13. ~~You'll need to find the URI string that connects Mongoose to mLab. Run this command to grab that string:~~
    * ~~heroku config | grep MONGODB\_URI~~
    * ~~Notice the value that appears after MONGODB\_URI =>. This is your URI string. Copy it to a document for safekeeping.~~
14. When you’re ready to connect Mongoose with your remote database, simply paste the URI string as the lone argument of your mongoose.connect() function. That’s it!
15. [Watch this demo of a possible submission](https://github.com/UCF-Coding-Boot-Camp/UCFLM-20170418-Class-Content/blob/master/Lessons/18-mongo-mongoose/Homework/Instructions/mongo-homework-demo.mov). See the deployed demo application [here](http://nyt-mongo-scraper.herokuapp.com/).
16. Your site doesn't need to match the demo's style, but feel free to attempt something similar if you'd like. Otherwise, just be creative!

**Instructions**

* Create an app that accomplishes the following:
  1. Whenever a user visits your site, the app should scrape stories from a news outlet of your choice and display them for the user. Each scraped article should be saved to your application database. At a minimum, the app should scrape and display the following information for each article:
     + Headline - the title of the article
     + Summary - a short summary of the article
     + URL - the url to the original article
     + Feel free to add more content to your database (photos, bylines, and so on).
  2. Users should also be able to leave comments on the articles displayed and revisit them later. The comments should be saved to the database as well and associated with their articles. Users should also be able to delete comments left on articles. All stored comments should be visible to every user.
* Beyond these requirements, be creative and have fun with this!

**Tips**

* Go back to Saturday's activities if you need a refresher on how to partner one model with another.
* Whenever you scrape a site for stories, make sure an article isn't already represented in your database before saving it; we don't want duplicates.
* Don't just clear out your database and populate it with scraped articles whenever a user accesses your site.
  + If your app deletes stories every time someone visits, your users won't be able to see any comments except the ones that they post.

**Helpful Links**

* [MongoDB Documentation](https://docs.mongodb.com/manual/)
* [Mongoose Documentation](http://mongoosejs.com/docs/api.html)
* [Cheerio Documentation](https://github.com/cheeriojs/cheerio)

**Minimum Requirements**

Attempt to complete homework assignment as described in instructions. If unable to complete certain portions, please pseudocode these portions to describe what remains to be completed.

**Hosting on Heroku**

Now that we have a backend to our applications, we use Heroku for hosting. Please note that while **Heroku is free**, it will request credit card information if you have more than 5 applications at a time or are adding a database.

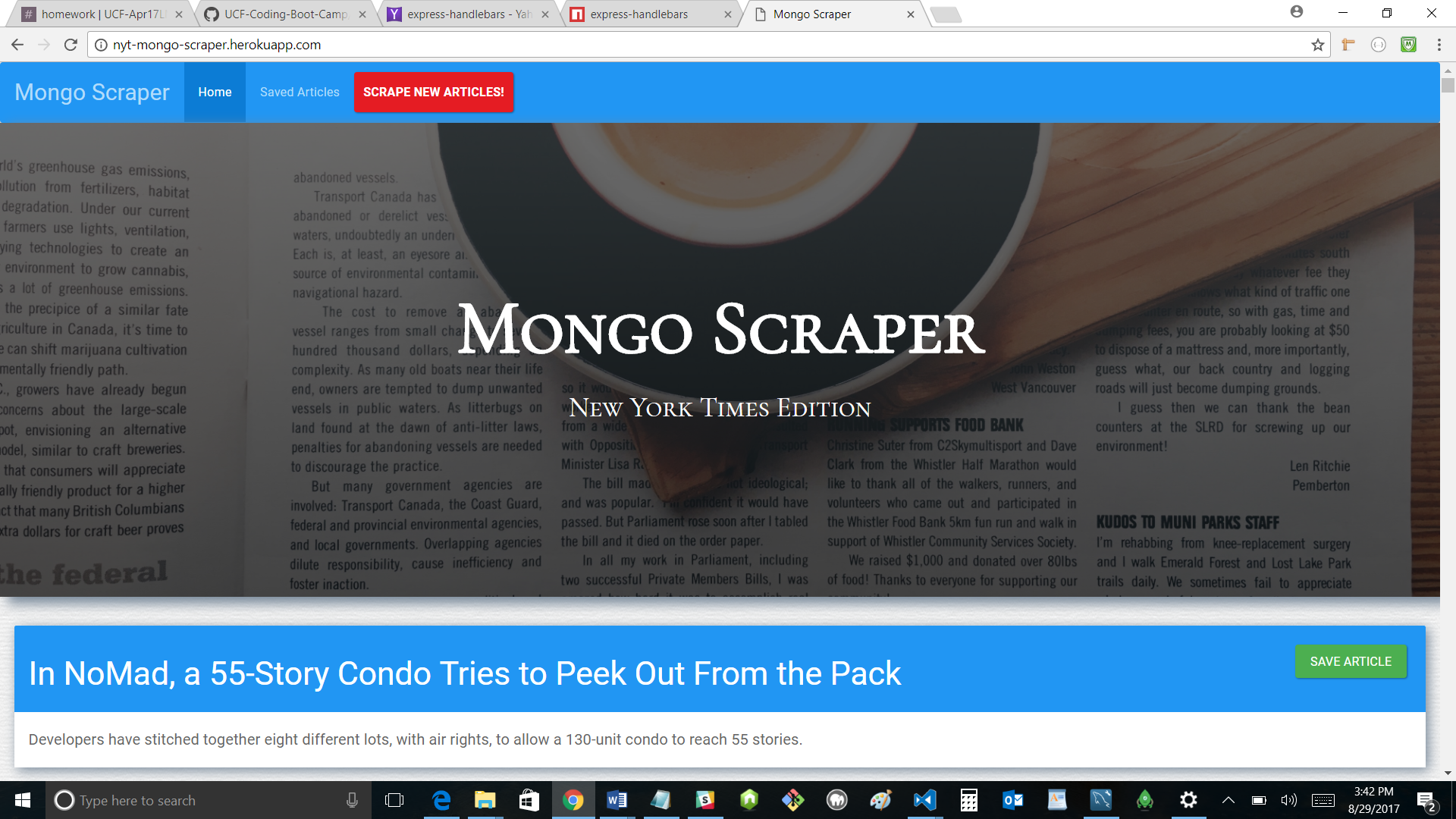
Please see [Heroku’s Account Verification Information](https://devcenter.heroku.com/articles/account-verification) for more details.

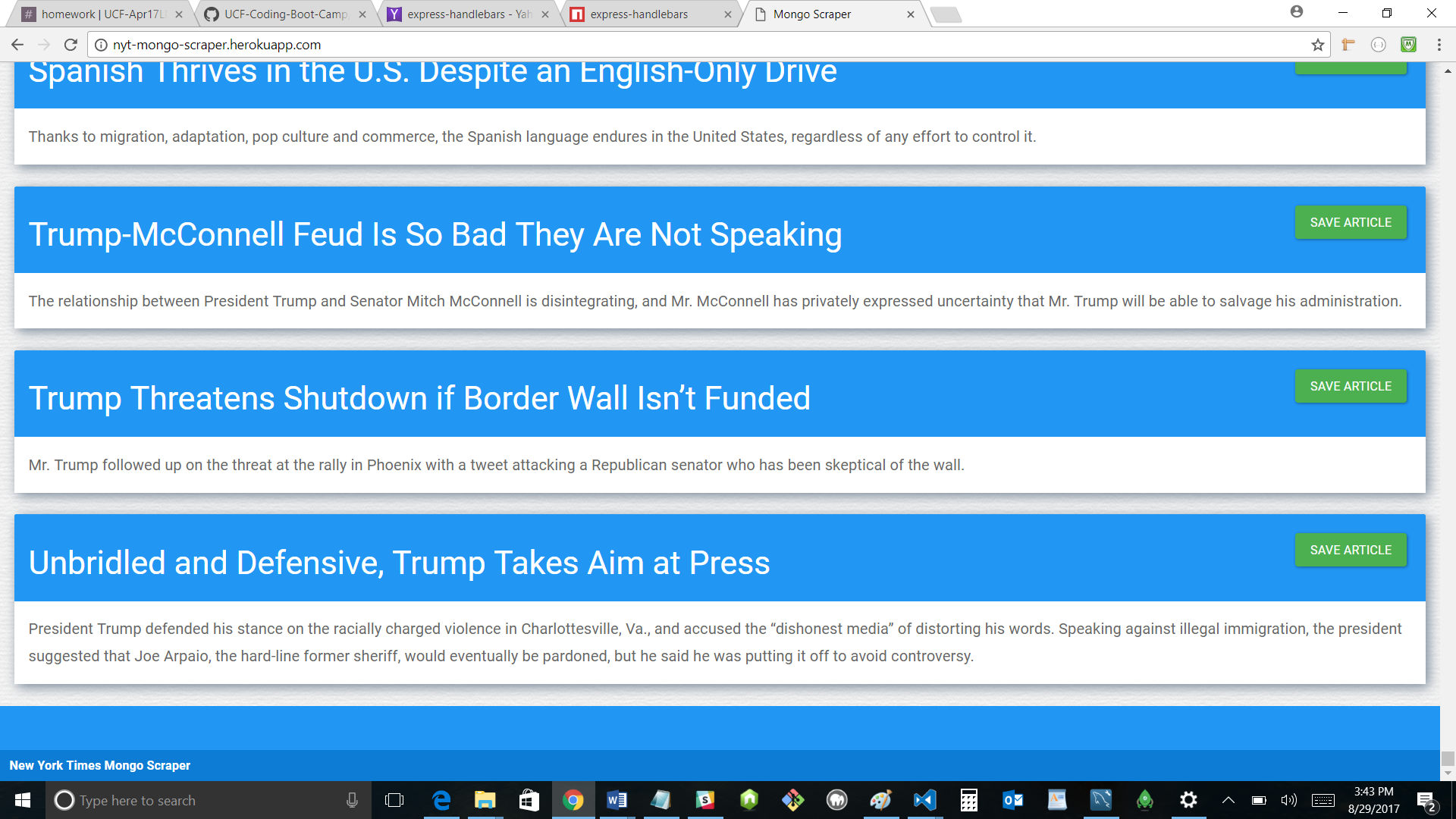
**One Last Thing**

If you have any questions about this project or the material we have covered, please post them in the community channels in slack so that your fellow developers can help you! If you're still having trouble, you can come to office hours for assistance from your instructor and TAs.

That goes threefold for this week: MongoDB and Mongoose compose a challenging data management system. If there's anything you find confusing about these technologies, don't hesitate to speak with someone from the Bootcamp team.

**Good Luck!**





==============

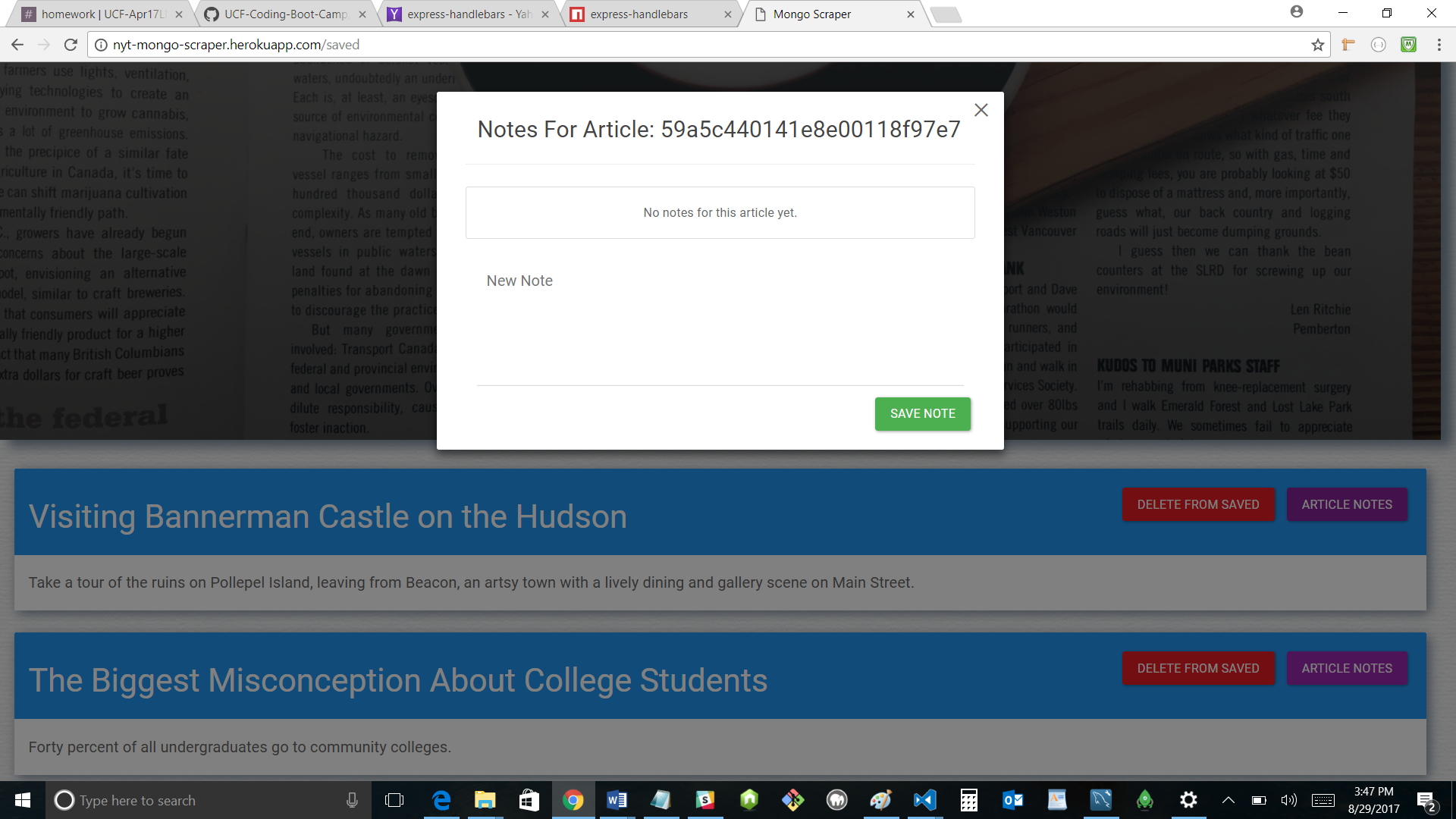
When “Scrape New Articles” button is pushed:



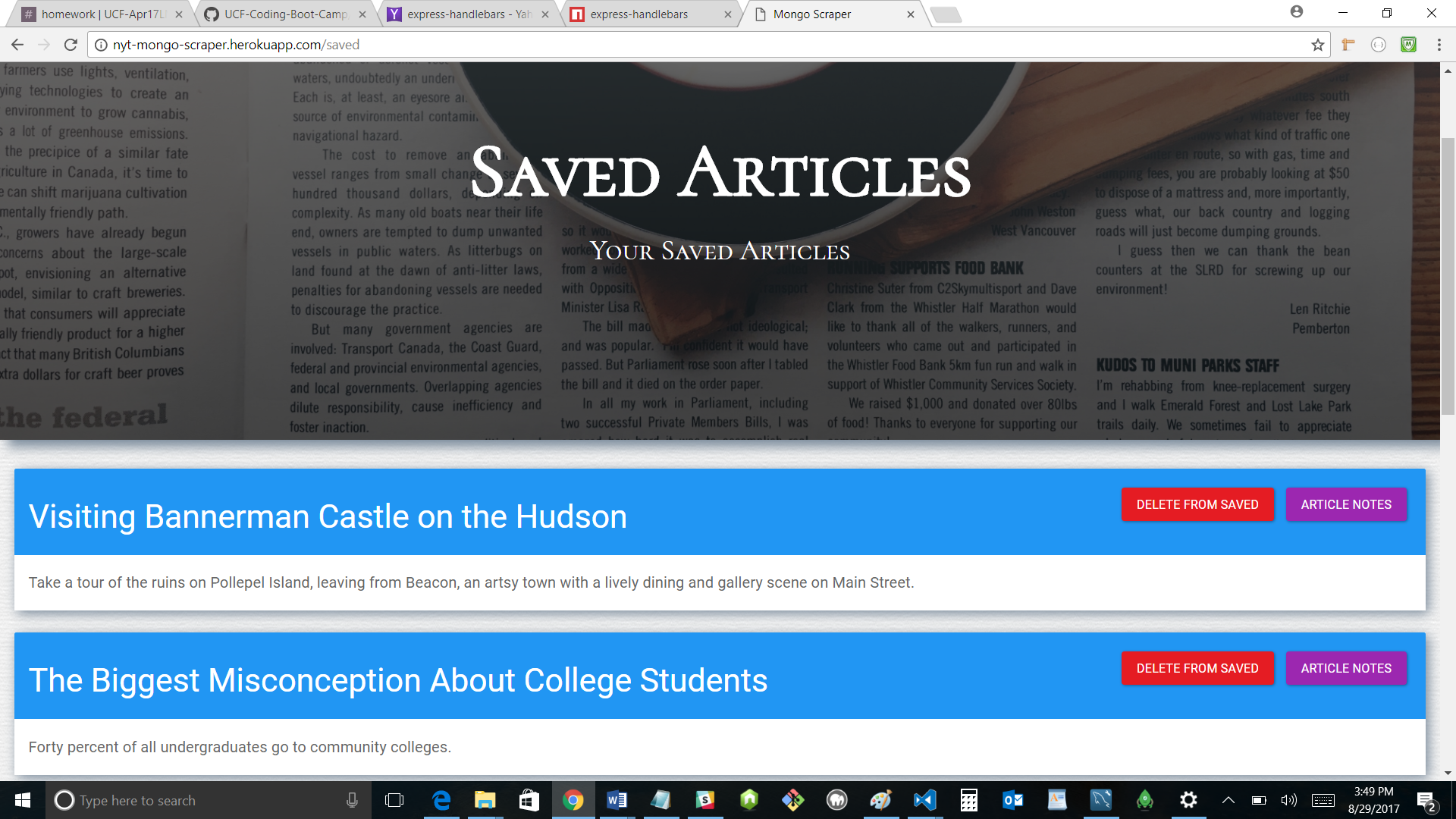
When “Saved Articles” link at top is clicked



When “Article Notes” button is pushed



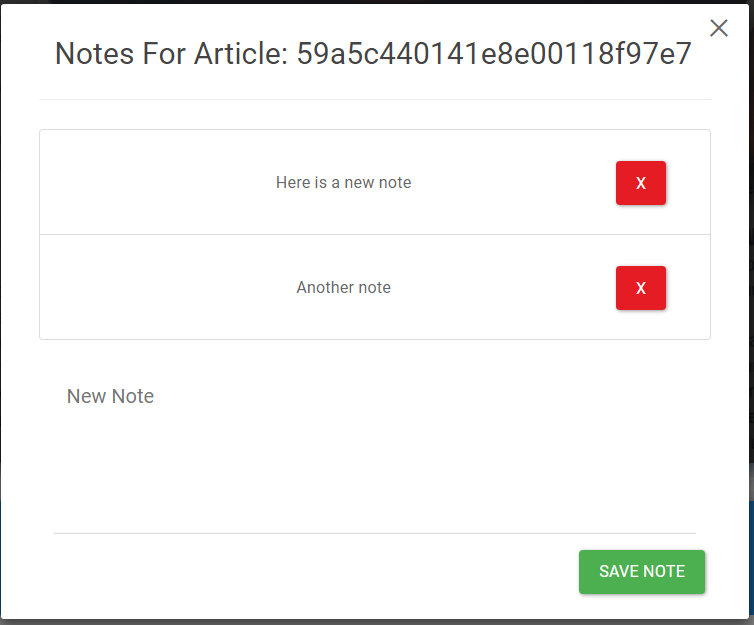
When a note is entered into textbox and “Save Note” button is pushed (pop-up just goes away)



When “Article Notes” button is pushed again (after note is added)



After two notes are added:



When “Delete from Saved” button is clicked, the article just disappears from the saved articles list