**Week 18 Homework Assignment - All The News That Fit to Scrape**

**Overview**

In this assignment, you'll create a web app that lets users 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.

**Remember**

* You will be fully capable of doing this homework by the end of Saturday's class.

**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:
   * express
   * express-handlebars
   * mongoose
   * body-parser
   * cheerio
   * request
   * **NOTE**: If you want to earn complete credit for your work, you must use all six of these packages in your assignment.
3. 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:
   * Create a Heroku app in your project directory.
   * Run this command in your Terminal/Bash window:
     + heroku addons:create mongolab
     + This command will add the free mLab provision to your project.
   * 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.
   * 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!
4. [Watch this demo of a possible submission](https://github.com/RutgersCodingBootcamp/04-16-NB-Class-Content/blob/master/homework-assignments/week-18-mongo-db/Week-18-Homework-Video.mov).
   * 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 follows this user story:
  1. Whenever a user visits your site, the app will scrape stories from a news outlet of your choice. The data should at least include a link to the story and a headline, but feel free to add more content to your database (photos, bylines, and so on).
     + Use Cheerio to grab the site content and Mongoose to save it to your MongoDB database.
  2. All users can leave comments on the stories you collect. They should also be allowed to delete whatever comments they want removed. All stored comments should be visible to every user.
     + You'll need to use Mongoose's model system to associate comments with particular articles.

**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)

**One Last Thing**

Like always, contact either the instructor or one of your TAs for support if you find yourself stuck while working on this assignment. 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!**