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.

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, express-handlebars, mongoose, body-parser, cheerio, request
4. \*\*NOTE\*\*: If you want to earn complete credit for your work, you must use all six of these

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. When you go to connect your mongo database to mongoose, do so the following way:

```js

// If deployed, use the deployed database. Otherwise use the local mongoHeadlines database

var MONGODB\_URI = process.env.MONGODB\_URI || "mongodb://localhost/mongoHeadlines";

// Set mongoose to leverage built in JavaScript ES6 Promises

// Connect to the Mongo DB

mongoose.Promise = Promise;

mongoose.connect(MONGODB\_URI);

```

\* This code should connect mongoose to your remote mongolab database if deployed, but otherwise will connect to the local mongoHeadlines database on your computer.

14. [Watch this demo of a possible submission](mongo-homework-demo.mov). See the deployed demo application [here](http://nyt-mongo-scraper.herokuapp.com/).

15. 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!

Submission on BCS: Please submit both the deployed Heroku link to your homework AND the link to the Github Repository!

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)

### Reminder: Submission on BCS

\* Please submit both the deployed Heroku link to your homework AND the link to the Github Repository!

---

### 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 and adding a README.md are required for this homework. In addition, add this homework to your portfolio, more information can be found below.

---

### 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.

---

### Create a README.md

Add a `README.md` to your repository describing the project. Here are some resources for creating your `README.md`. Here are some resources to help you along the way:

\* [About READMEs](https://help.github.com/articles/about-readmes/)

\* [Mastering Markdown](https://guides.github.com/features/mastering-markdown/)

---

Add To Your Portfolio

After completing the homework please add the piece to your portfolio. Make sure to add a link to your updated portfolio in the comments section of your homework so the TAs can easily ensure you completed this step when they are grading the assignment. To receive an 'A' on any assignment, you must link to it from your portfolio.

---

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.