

CSCI 421 – Advanced Web Development

Lab 3 – Data Driven Website

For this lab you will start with your Blogger application from Lab 2 as the basis and modify it to include a database and model and refactor it to include controller(s) that send static data to the views. Lastly, you will publish this version of your application to your previously created GitHub repository as a new branch named: “Lab 3”.

DUE DATE: *This lab is due to the instructor by Monday, February 19th at 11:59pm.*

Instructions:

1. Using your Amazon Lightsail MEAN instance, make a copy of your previous Lab 2 application either using Linux copy commands or GIT commands to pull down the previous application to a new (different than previous) directory.
2. Your application will be similar to the application you wrote in lab 2, but will be updated to include a database connection, a data model, static data from a controller, etc.
3. Your web application must adhere to the following specification:

Specification:

- i. **Port 80:** Your application must load from a browser via port 80, which is the standard HTTP port for a server.
- ii. **Use an “app_server” directory:** Your program should be refactored such that all the folders (*routes, views, controllers, models*) holding the fundamental pieces of your application are found under the *app_server* directory.
- iii. **Create a Database:** Having read part “2:5 - Building a Data Model with MongoDB and Mongoose” you will be armed with the information and know how in order to:
 - a) Install Mongoose.
 - b) Create a new user account (with password) for you and your application to use when connecting to the database. I suggest calling this the “blogs” user.
 - c) Create a ***blogs.js*** model file that includes a *blogSchema* schema for a blog entry. At a minimum, a blog entry should have a blog-title, blog-text and a created-on date.
 - d) Author and integrate ***db.js*** database file.

- e) **Update *app.js*** to load your *db.js* file. Run your application in debug mode and make sure it is writing to the console that the database is connected properly when starting.

 - iv. **Update blog data controller:** Update your *blog.js* (or whatever you named it) controller file such that it returns a listing of at least 3 blog entries that conform to the schema definition defined earlier.

 - v. **Update blog list view:** Update your blog listing page view so that it displays the static data from a controller in a nicely formatted list. Add icon links to blog edit and blog delete for each blog entry in the list page.

 - vi. **Create Blog Edit and Blog List views:** Author blog editing and blog deletion views and indicate these features are under construction, then wire them into your application by updating *index.js* router, and your *blogs.js* controller as needed.
4. Your application must be setup to run even with your MEAN instance is not connected.
 5. Using your GitHub account and the repository created and used prior for Lab 2, save this lab as a new branch (“Lab 3”).
 6. Below is the instructor’s example:

<http://3.143.17.202/>

7. Get your application working by developing database, models and updating and adding to routers, controllers, views and updating *app.js* as required.

8. Once you have the application working and it is available via port 80 even when disconnected from your MEAN instance and you have your app uploaded to your GitHub repo, please send the full URL of your app AND your GitHub repo to the instructor via email with subject of “Lab3”. Important:

<mailto:thomas.rogers@millersville.edu?subject=Lab3>