

CSCI 421 – Advanced Web Development

Lab 2 – Node and Express Site in GitHub

For this lab you must create a MEAN project from the ground up according to a simple specification of the user experience and technical requirements. This lab will be slightly different than the previous in its construction and you must also place this lab in your own GitHub repository.

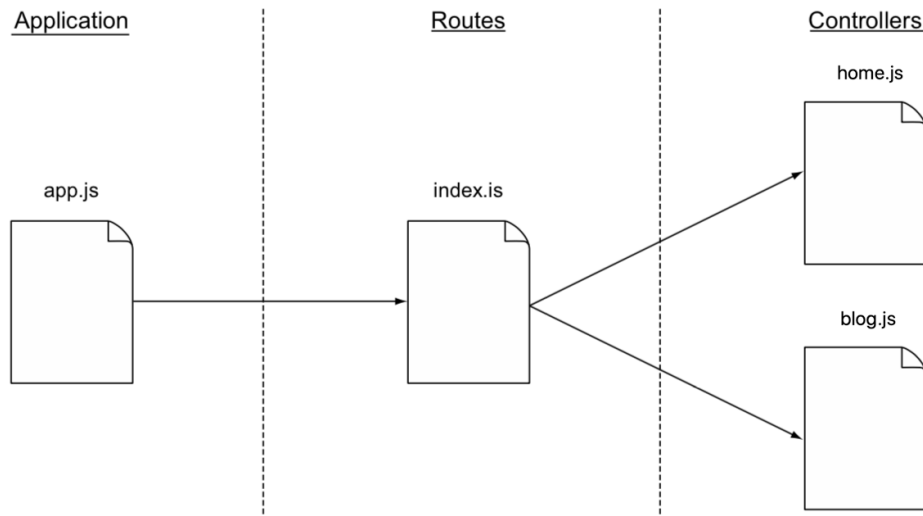
DUE DATE: *This lab is due to the instructor by Monday, February 12th at 11:59pm.*

Instructions:

1. Using your Amazon Lightsail MEAN instance, develop a new application (suggest you call it “blogger”).
2. Your application will be similar to the application you wrote in lab 1, but will be built using a “Routes and Controllers” architecture (Manning Part 2:4, also covered in Lecture 3.)
3. Your web application will start as standard Express application that adheres 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. **“Routes and Controllers Architecture”:** Your program must be architected using an index.js file as its only routes file, and all the other routing will be done in controller files as shown in the following diagram:



- iii. **Home page:** The home (aka default) page will be driven by a view that renders a page as follows:
- Title** – The HTML **title** of the page shall be: *<your name> Blog Site*
 - H1** – The H1 tag of the page shall also be: *<your name> Blog Site*
 - First paragraph** – The first paragraph of the page should be an introduction.
- iv. **“Blog List” and “Blog Add” pages:** Your app shall also include blog pages that for now will each indicate that the blog features are under construction (you will work on these features in future labs).
- Title** – The HTML titles of the pages shall be “Blog List” and “Blog Add” for the listing and add pages, respectively.
 - H1** – The H1 tag of the pages shall be the same as their titles.
 - First paragraph** – Simply put some type of “under construction” message here.
- Note:** You should only need one controller file, *blog.js*, for all your blog related pages. The book calls this a “collection of controllers”. Within *blog.js* there will be controllers that route to each of the different views for your “Blog List” and “Blog Add” pages.
- v. **Navigation Menu:** Your program must include some type of navigation menu at the top of each page, with three navigation choices, as follows:
- My Blog** – Navigates to the home page.
 - List Blogs** – Navigates to the blog listing page.
 - Add Blog** – Navigates to the blog add page.

- Note:** It is recommended you take a look at the “todo” app from the Lab Intro exercise for an example that uses **bootstrap** for its navigation menu. Also, remember, when adding dependencies to your app you have to update *package.json*, re-run “npm install” to install the dependencies, and typically update *app.js* in order to add virtual paths to resources (like .css and .js files) under the app’s node_modules directory in order for your application to find said resources.
- vi. **App Runs when Disconnected:** Your application must be setup to run even with your MEAN instance is not connected.
 - vii. **Place your app in GitHub:** You must create a GitHub account, if needed, and create a repo for your application and upload your application to the repo. See Lecture 3 notes for details on this.
 - viii. **Sample:** Below is the instructor’s example:

<http://3.143.17.202/>
 - ix. Get your application working by developing and updating the needed router, controllers, views and updating app.js as required.
 - x. 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 “Lab2”. Important:

<mailto:thomas.rogers@millersville.edu?subject=Lab2>