# HTTP5222 Assignment 1 (Node.js Express app)

Create a Node.js website (with Express and Pug—or a different template engine if you want to try a different one) containing **at least** three public pages (not including any admin pages). This should be for a dummy website for a small business (come up with your own fake business). Add MongoDB as the data source for some of the content (e.g. a MongoDB collection for product data). You only need one collection to demonstrate your understanding (**do not use the same collection as was used in the in-class example**).

**Requirements:**

1. Create a website using Node.js with Express and Pug (or a different template engine) for a fake small business (preferably, not the same as in Lab 2).
2. There should be at least three public pages (home page plus two more).
3. Create a MongoDB collection to store data for some of the content.
   * For example, if it's a shop, create a collection to store product data (use this for a list of products). If it's an animal shelter, make a collection to store data about animals at the shelter. (You don't need to create a collection for everything—just the main listable data.) If it's a hotel site, you can create a collection for hotel room data.
   * Use this collection as your data source (on public page **read**).
4. Add a create page for easy data-adding (no login required for this assignment due to relatively short turn-around time though it’s a nice to have if you have the time).
5. Your MongoDB collection should be online (e.g. on MongoDB Atlas) so that I can see the data in your app when I run the app code. (Make sure to make the DB data **available from anywhere**.)
6. Style the page to look like a realistic website.
   * You may use a CSS framework but change colours and fonts. (It shouldn't be obvious which framework was used at first glance.)
   * It should be responsive.
7. Flesh out the content (realistic content only—**do not use lorem ipsum**). **Cite your image source URLs in comments or a README!**

You will be marked on:

* **Code quality (10)**
* **Design/Usability (5)**

This assignment is **individual** and is worth 20% of your final grade. You can put your web app on GitHub and just submit your GitHub link (.gitignore the ***node\_modules*** folder).

Due date: **June 14, 2024 @ 11:59pm**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Excellent** | **Good** | **Satisfactory** | **Somewhat Unsatisfactory** | **Unsatisfactory** |
| **Code Quality (/10)** | Code works, is efficient, well-commented and formatted. Code follows best practices. | Code works, is commented and formatted well. Code mostly follows best practices but for a few minor issues. | Code is a little buggy, is not always commented or formatted well but a good attempt. Code not always efficient. | Code is buggy and poorly commented. Some issues following best practices. | Code doesn't work and is not commented. Code has many major issues. |
| **Design  (/5)** | Design is professional. Website is easy to use and follow and is accessible. | Design is mostly professional but with a few minor issues. Website is easy to use and follow but with some minor issues. | A good attempt but feels like student work. Website is mostly fine but with many minor issues or at most one or two major issues. | Design is not really complete and is obvious it's student work. Website has some major issues around usability/accessibility. | Design not really fleshed out at all. Design is not complete. Website is confusing/hard to use. Website not accessible. |