

CPSC 2261: Web Technology

Summer 2023

Assignment 9: The Local Library Tutorial: MVC application using Node/Express.js

Learning goals:

- Use the Express Application Generator tool to create a skeleton website and application.
 - Learn how to connect your Express app to **MongoDB Atlas** using Mongoose
 - Create routes for requesting different information, and templates to render the data as HTML to be displayed in the browser.
 - Write **forms** in Express using Pug to **create**, **update** and **delete** documents.
 - Use the **express-validator** module to perform both **validation** and **sanitization** of our form data.
 - Get web app ready for production and **deploy it to Railyway**
-

Notes:

- This is an **optional assignment**. I will count your best 8/9 assignments towards your final grade calculation.
 - Heads up, this is a long tutorial. However, the concepts you learn along the way will be worth the time invested if you want to gain experience with server-side development and want to have a thorough understanding of the MVC design pattern.
-

Setup:

- This will be a **tutorial style** assignment, where you will follow along the following MDN Express tutorial article series (Parts 1-7):
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Tutorial_local_library_website
- This will introduce the fundamental concepts of working with Express and MongoDB Atlas. You will learn how to work with HTML Forms in Express using Pug.

Task 1: Creating a skeleton website:

In this task, we will create a "skeleton" website using the Express Application Generator tool, which you can then populate with site-specific routes, views/templates, and database calls. In this case, we'll use the tool to create the framework for our Local Library website, to which we'll later add all the other code needed by the site.

- Read the following article from MDN Docs:
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/skeleton_website
- **Follow the steps outlined in the article above** (Part 2) to create a skeleton website using the Express Application Generator.

Task 2: Using a Database (with Mongoose)

Now that we have our skeleton Express application set up, we will look at databases, and how to use them with Node/Express apps. Specifically, we will use [Mongoose](#) to provide database access for the LocalLibrary website. Along the way, we will set up a free [MongoDB Atlas](#) account. Additionally, we will learn how object schema and models are declared and look at a few of the main ways in which you can access model data.

- Read the following article from MDN Docs:
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose
- Follow the steps outlined in the article above (Part 3) to connect your application to MongoDB atlas.

Task 3: Routes and controllers

In this section we'll set up routes (URL handling code) with "dummy" handler functions for all the resource endpoints that we'll eventually need in the LocalLibrary website. On completion, we will have a modular structure for our route handling code.

- Read the following article from MDN Docs:
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes
- Follow the steps outlined in the article above (Part 4) to create all the routes for the site

Task 4: Displaying library data

We're now ready to add the pages that display the LocalLibrary website books and other data. The pages will include a home page that displays how many records we have of each model type and list and detail pages for all of our models. Along the way, we'll gain practical experience in retrieving records from the database, and using templates.

- Read the following article from MDN Docs:
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Displaying_data
- Follow the steps outlined in the article above (Part 5) to display the library data

Note: You are **not required** to share your login credentials for your primary MongoDB account. You can create a separate MongoDB atlas account that you plan to use for this course only.

Task 5: Form handling, validation and sanitization

We will now learn how to work with HTML Forms in Express using Pug. We'll extend the LocalLibrary website to allow users to *create, edit and delete* items from the library.

- Read the following article from MDN Docs (Part 6):
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/forms
- Follow the steps outlined in the article above to complete the “Working with Forms” section.
- Note that there is a **challenge** section that asks you to implement the delete pages for the Book, BookInstance, and Genre models, linking them from the associated detail pages in the same way as our Author delete page.

Task 6: Deploy to production

Now that we have created the LocalLibrary website, we will deploy it to a service called Railway

- Follow the steps outlined in the article below to complete the “Deploying to production” section.
https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/deployment

Hand In:

Zip your assignment directory and submit your work in the Assignment 9 folder.

When you zip your directory, remember to exclude the node modules folder. The marker will download your submission, install the node modules and run your app.

Checklist:

- [4 marks] Task 1 + 2: Create skeleton website + connect to MongoDB
- [4 marks] Task 3 + 4: Routes, Controllers + display data
- [5 marks] Task 5: Form handling, validation and sanitization
- [2 marks] Task 6: Site deployed to Railway (remember to include the URL)

Total: 15 marks

Reference(s):

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Tutorial_local_library_website