

# Assignment 1

## COMP 3015: Web Application Development

### Your mission:

Design and implement a news aggregation web application (think of a basic version of [HackerNews](#) with only the core article related functionality – no commenting functionality required).

The front page of your application will display a listing of all submitted articles. A navigation bar should be created with buttons that can be used to navigate to an article submission page. The article submission page will have a form with inputs for the article title and associated link. Edit and deletion functionality must also be implemented.

### Requirements:

- A user will need to be able to submit a link to an article as well as a title for the post. This data should be written to a file in a format that can later be read.
  - I advise you to use a JSON format as we did for lab 2.
- Validation must be performed to ensure a proper link is submitted, as well as a title that is not empty.
- A user must be able to edit the title and link after submission.
- A user must be able to click on a given article link on the index page, and get to the linked resource (in other words, submitted links remain usable).
- A user must be able to delete a submitted post.
- A user must be able to view all submitted articles on the index page.
  - This is the page where you will need to read the file that contains all of the article submission data.
- All code must be reasonably formatted and use reasonable naming conventions.
- All Create, Read, Update, Delete (CRUD) operations must be implemented using PHP. No JavaScript should be needed for this assignment.
- The system you build must be able to handle multiple posts with the same title and link
  - **Idea:** give each article an ID. You could generate this ID using [time\(\)](#) or similar.
    - For the purposes of our assignment, this is good enough since we won't have multiple posts being made at the same second.
    - The user should **NOT** have to manually enter an ID in order to view/delete/update articles.
- Style the application in order to make it reasonably presentable to end users.
- Provide a README.md file containing information on how to run your application, and any setup or configuration needed (for example, you might state that a "data/articles.json" file must exist before running the application).

Note: CSS frameworks such as [TailwindCSS](#), and component libraries such as [DaisyUI](#) or [TailwindUI](#) make it easier to develop nice looking applications. Check these out!

A starter kit has been provided on D2L, but you can also start from scratch if you like.

### **Submission:**

- Commit and push your changes to your private GitHub repository and submit the link to the D2L dropbox when completed.
- Create a screen recording where you do a quick walk through of your application
  - Show the JSON file after create, update, delete operations

### **Evaluation:**

- Application Functionality: / 80
  - Article submission / 20
  - Article update / 20
  - Article deletion / 20
  - Index page article list / 20
- Code Readability / 10
  - You must have reasonable indentation, variable names, etc. for full marks.
- Docs (README.md) + screen recording / 10
  - Explain how you run your application, any limitations it has, assumptions you made during the assignment, etc.

Total / 100