

## Assignment 2

### COMP 3015: Web Application Development

#### Your mission:

Update your news aggregation site to use a database in order to store articles, and add functionality to support multiple users. A user should only be able to delete or update article submissions that they themselves have made. In order to determine if a user has submitted an article, we will build an authentication system (functionality to register a user, login, logout). This will allow us to associate a submitted article with a particular user record in the database. Unauthenticated users will not be able to create, update or delete articles, but will be able to view submitted articles from other users.

#### Requirements:

- IDs do not need to be manually entered by a user.
  - All IDs should be in URLs, [hidden form inputs](#), or similar.
- Two MySQL tables: **articles** and **users**
  - A database schema is given as a “schema.sql” file. You can run it in order to generate your database and tables
- A user must be able to register for an account with a name, email, password
  - Validate that the password is > 8 characters and contains at least one symbol.
  - Check the email is valid (is there a way to do this without explicitly using regex?)
  - The stored password **must** be a digest of a cryptographic password hashing function (use [password hash](#) for this and use Bcrypt with a cost of 12)
- A login page with a form accepting an email and password
  - The provided login credentials must be checked against the data stored in the **users** table. Note that you will need to use [password verify](#) to check that the plaintext password matches the stored digest
- Authenticated users should have a session with their user ID in it
  - Unauthenticated users can still have a session, but there won't be a session ID in it
- A user must be able to log out of the application
  - This destroys the session and all associated data
- A user must only be able to update and delete submissions that they themselves have made (see marking, this is “Authorization”)
- Queries made to the database must not be vulnerable to SQL injection attacks
  - Parameterized queries must be used
- All article and user CRUD operations must use a database (not a file as we did in A1)
- Style the user interface of the application to make it presentable

- The session cookie must be HttpOnly (no code changes needed, just a php.ini file change)

A starter kit has been provided on D2L.

### Submission:

- Submit a ~1 minute screen recording going through the functionality of the application and proving each part works.
- Provide all of the files needed to run your web application as well as a README.md file that contains instructions on how your program can be run.

Images for reference (note that your application does **not** need to look exactly the same):

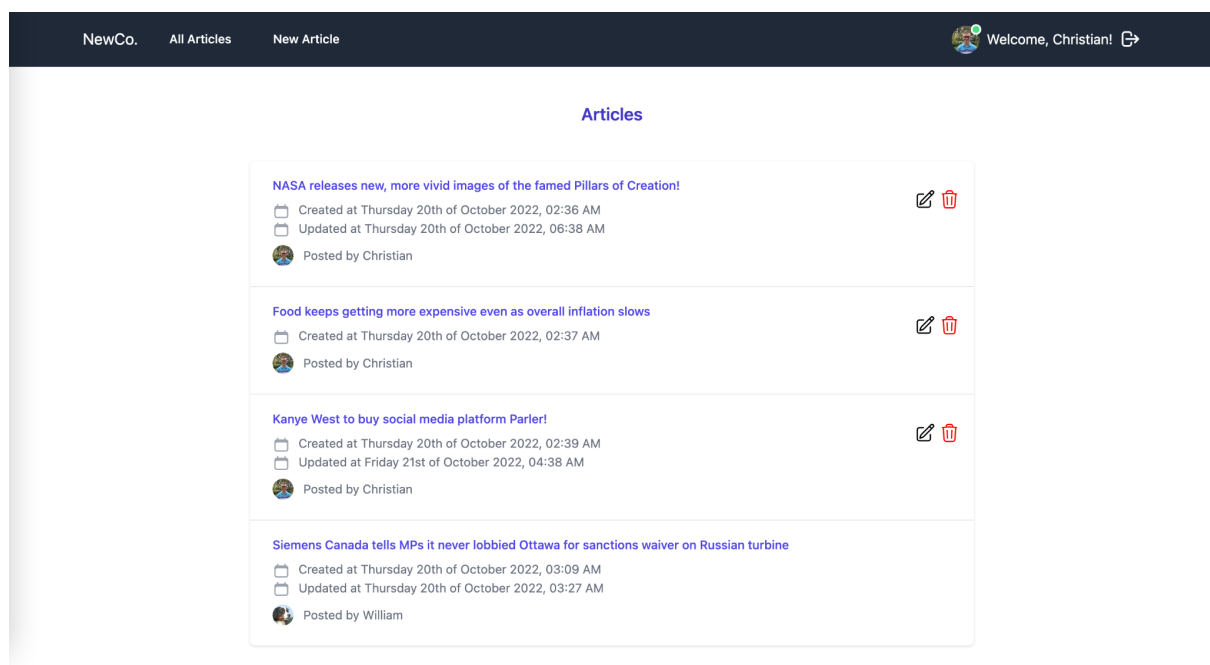


Fig 1. The articles index page. Note that the first three posts are created by the currently authenticated user, so they can be edited and deleted. The fourth post was created by a different user so it cannot be edited or deleted by us.

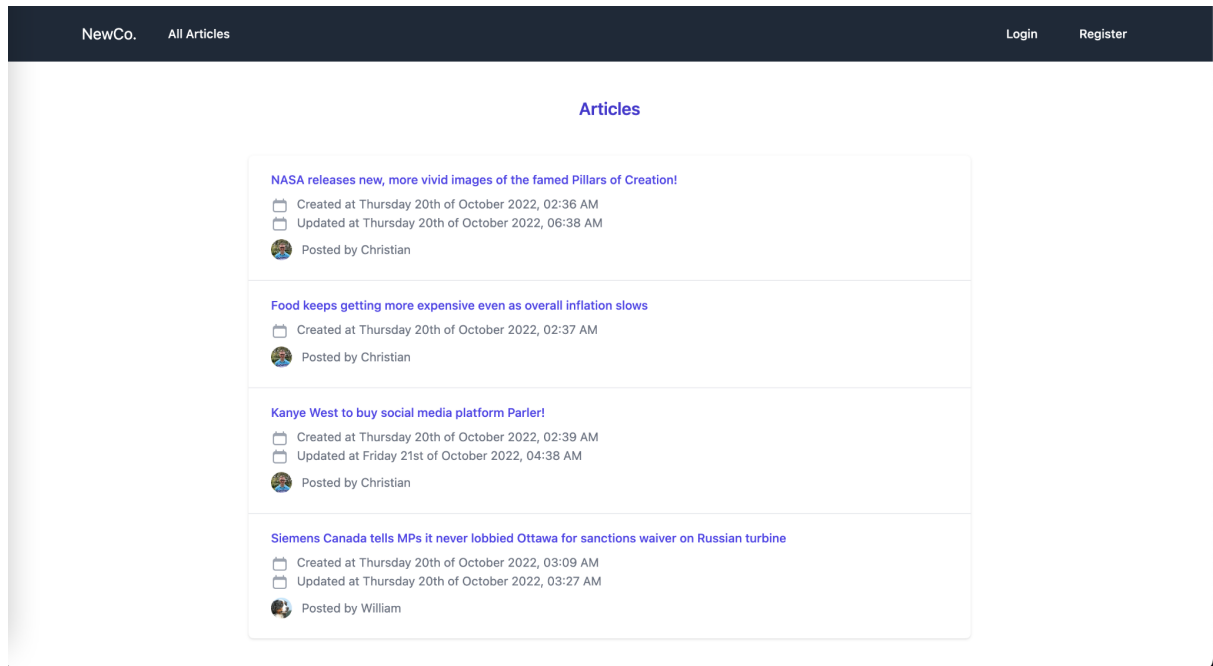


Fig 2. An unauthenticated user can view the index page but cannot edit or delete anything. Note that the title bar changes to show login and register links.

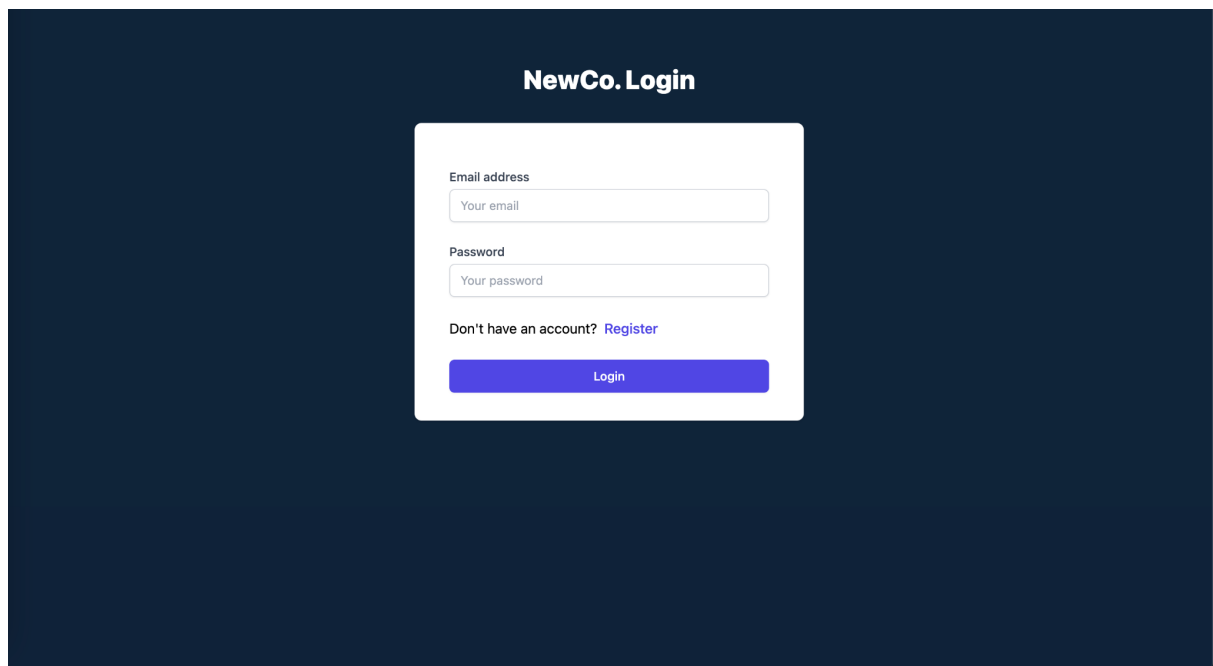
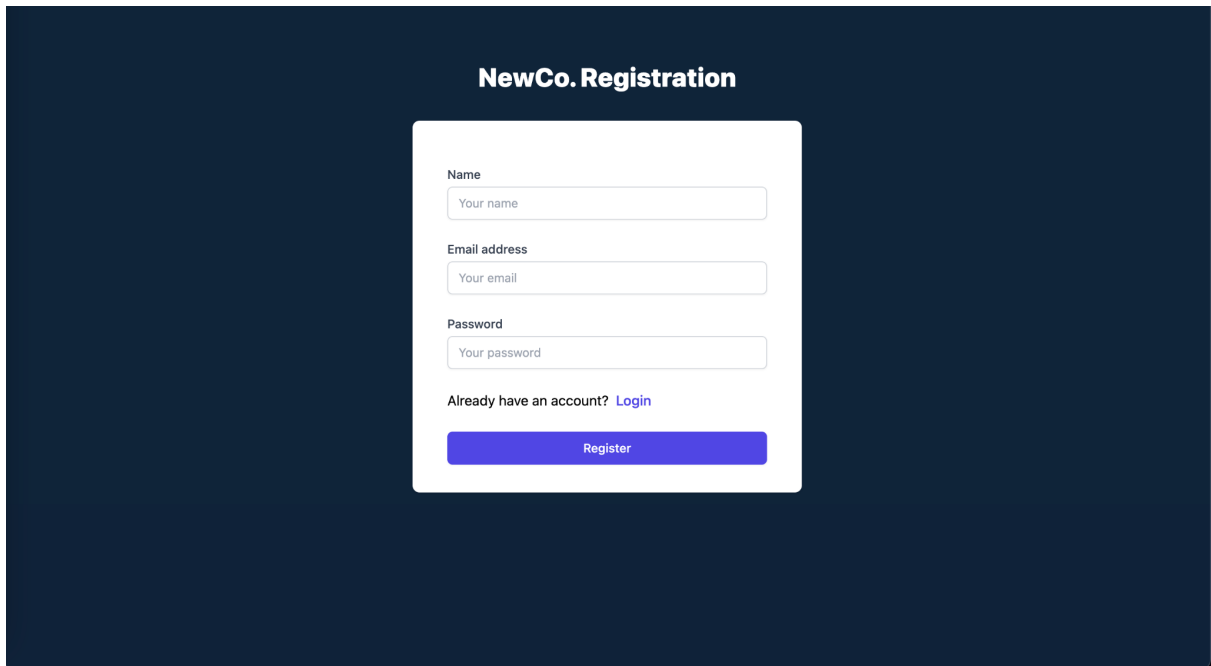
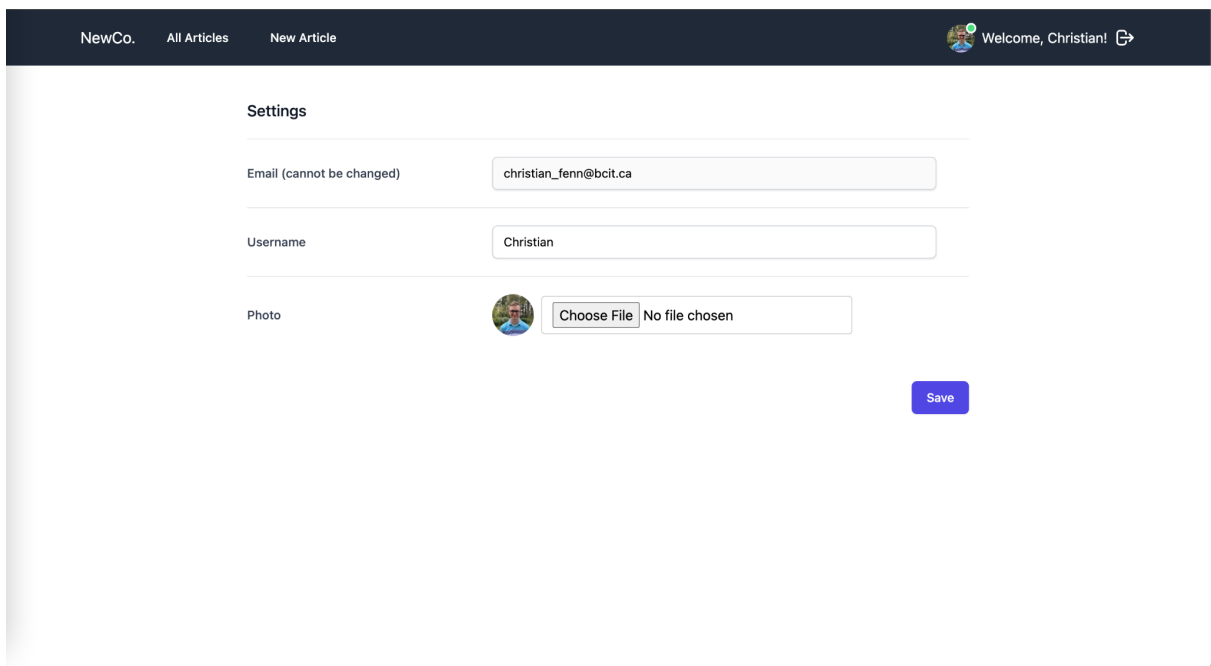


Fig 3. The login page



The registration page features a dark blue background with a white registration form in the center. The form is titled "NewCo. Registration" and contains three input fields: "Name" with placeholder text "Your name", "Email address" with placeholder text "Your email", and "Password" with placeholder text "Your password". Below these fields is a link "Already have an account? Login" and a blue "Register" button.

Fig 4. The registration page



The settings page has a dark header with navigation links "NewCo.", "All Articles", and "New Article" on the left, and a user profile section on the right showing a profile picture, the text "Welcome, Christian!", and an edit icon. The main content area is titled "Settings" and contains three sections: "Email (cannot be changed)" with the value "christian\_fenn@bcit.ca", "Username" with the value "Christian", and "Photo" with a profile picture and a "Choose File" button. A blue "Save" button is located at the bottom right.

Fig 5. The Settings page (accessible by clicking on the photo of the user)

### **Evaluation:**

- Application Functionality: / 90
  - Registration / 10
  - Login / 10
  - Profile picture upload / 10
  - Username update / 5
  - Logout / 5
  - Article submission / 10
  - Article update / 10
  - Article deletion / 10
  - Index page article list / 10
  - Authorization / 10
- Application styling / 5
- Code style, readability, documentation / 5

Total / 100

Due at the start of week 9.