Project Title: Learning Path Creator

Completion date: December 4, 2023

Table of Contents

Contents

Introduction

Briefly describe the purpose and functionality of the application.
The purpose of the application is to host a community of learning-driven and educational members on a web-based platform. The primary purpose of application is to provide users with a space here they can create, share, and explore curated lists of educational resources.

Setup and Installation

Provide step-by-step instructions on how to set up and run the application.

1. Visit the website
2. Click ‘Sign Up’ to register an account
3. Choose a username and password
4. Enter your email
5. Click ‘Sign Up’
6. Scroll the Home page
7. Select where you wish to proceed

a. Explore the extensive knowledge available in Learning Paths and Categories
b. Look at your Profile
c. Get expert certified (bugged)
d. ~~Create your own Learning Path or Resource (bugged)~~
e. Search for a specific title or description
f. ~~Vote on learning paths and resources~~

Application Architecture

Describe the overall architecture of the application, including any frameworks or libraries used.
At the beginning the development process, a member of the team created a detailed Entity Relationship Diagram (ERD) to outline the project needs. Through som
e mutual trial and error analysis, we got the final framework used. That same member created the database used to store the information provided by users. That information includes – but is not limited to -; learning paths; resources; user data. The database was uploaded to PHPMyAdmin to get our php back-end to work. The php back-end was created by our second team member. They created an object oriented approach to data collection and distribution. Together, our members put the database and back-end together with borrowed CSS and simple HTML to create the user interface. Overall, the application architecture contains web- and server-side, user interface, authentication, database and user management elements.

User Authentication and Authorization

Detail how user authentication and authorization are handled in the application.
Include a few usernames and passwords.
The user authentication and authorization are handled by both a database and object oriented php. The database contains the stored user information while the php contains the update-able objects and connection to the database. For security reasons, all of the provided passwords are hashed by the php code. In the user interface, there are two pages which include the authentication and authorization code; login and sign up. Both pages require connection to the database through php.

Database Interaction

Explain what method your application uses to interact with the database. Insert the picture of your database schema.

Database Queries

Please include all the queries (with a short comment about where it is used) you used within your project.

Server-side scripting (PHP)

The utilization of HTML forms and PHP were vital in the server-side scripting process. The post method was a focus in the sign-up, login and search processes to collect the users input data from a form. While the session was started when the user opened the page, it wasn't used until they logged in or signed up. Upon logging in, the session holds the users username to provide personalized pages via the get method. The get method was used dynamic profile, categories and learning path pages to display either all information or only information for a specific category/learning path. In the data handling, we used OOP methodologies with the information kept in the database. The magic getters were used inside foreach loops when displaying information from the object arrays.

Responsive User Interface

Explain how responsiveness is achieved in the application's user interface.
The responsive user interface was copied from the free code available from w3schools https://www.w3schools.com/css/css_rwd_templates.asp . They used media queries to fit the content to the correct page size, flexible display to design the content backgrounds, and relative units when displaying elements. Minor edits were made to fit the application specifications, such as adding a cancel button where the nav bar does not exist.

Error Handling and Validation

Detail the error handling and validation mechanisms in place throughout the application.
Validation mainly comes from the Session variables. When a user is logged in, the session holds their username which updates the nav bar and available web pages. Unfortunately, our error handling was minimal upon completion as some unforeseen issues arose during the development process. What error handling we have is mostly through the HTML required keyword in forms and basic handling in the database manager script.

Documentation and Code Quality

Discuss the measures taken to ensure code quality and provide additional documentation if necessary.
At the beginning of the project, a member of the team created a GitHub repository and Discord to share files and communicate between the team. The GitHub repository aided our members by sharing the base scripts they were creating, as one member was working with the database and another with php. The discord server helped communication, which improved the code quality because we were able to discuss any elements we wished to change, or had better ideas for, without having to schedule time to call. There is not a lot of documentation, as we did not manage our time well enough to complete the project on time.

Project-Specific Functionality (Innovative Feature)

List and describe all the project-specific functionalities implemented in the application.
The innovative feature we attempted to add was an Expert Validation element. The idea was that a user could share documentation (such as a degree, certificate or job title) stating that they were an expert in a specific field. This would translate over to Learning Paths and Resources that they created or up-voted, to inform other users that the content was created/endorsed by an

expert. There was supposed to have been a class that took the information to validate the users experience. Alas, our hearts were bigger than our motivations. The implemented section of this is a Boolean variable within the User class that could be passed along to a Learning Path created by the user. While a new user couldn't validate themselves, the users already populated in the database could.

Additional Notes

Include any additional information or notes that might help the reader to understand or use the application.

Team Member Information and Contributions

Provide contact information for all group members, including names, student IDs, email addresses, URL to presentation video and URL project folder on member's GBLearn account.

All URLs must open in a new window.

Full Name: Nicole Milmine Student ID: 101462077 GBC email: Nicole.milmine@gerogebrown.ca	
Video URL	https://vimeo.com/891166717?share=copy
GBLearn URL	https://f3462077.gblearn.com/comp1230/assignments/project/
Tasks completed by member	Creator of the GitHub Repository
	Input the ERD blueprint into the Back-end code - Just base stuff like creating files, classes, members, constructors and method skeletons each filled with comments on how it should work and what it should reference. Literally outlined exactly what we're going to do Updated the framework from the original ERD to one that made a bit more sense (ex. Moving the Create_Resource method to the resource class)
	Discussed UI design with Pablo
	Discussed Pablo's Expert Certification and Vote idea for the innovative feature Vote did not make it into the final draft
	Back-end OOP framework
	Completed most information in the files; Create_Resource.php, Home.php, Categories.php, Categories_page.php, Resource_page.php, login.php, logout.php, signup.php, Learning_Path_Page.php, Profile.php User.php, Learning_Path.php, Resource.php
	I took the css from w3schools
	Took the arrays with database information and created object arrays to be used throughout the application (See Main.php)
	Helped Pablo with the README file

Full Name: Pablo Arango Gomez Student ID: 101153741 GBC email: pablo.arangogomez@georgebrown.ca	
Video URL	https://vimeo.com/manage/videos/891282082
GBLearn URL	https://f3153741.gblearn.com/comp1230/assignments/project/
Tasks completed by member	Sorted out why our code wasn't working once uploaded to GBLearn – it was because we were using a newer version of php
	Database creation and implementation - SQL to create the database - SQL to fill the database - DatabaseManager class that connects to and takes information from the database - config.php file to connect to the database and appropriate server(s)
	Created the ERD, Use-Case Diagram, & Class Diagram
	Created the Search Portal
	Thought up & implemented the Innovative feature
	Discussed UI Design with Nicole
	Discussed the benefits of constant connection to the database vs OOP methods and occasional connection
	Helped Nicole understand the assignment rubric and how the project should turn out
	Debugged many of Nicole's files (Create_Learning_Path.php specifically)
	Implemented user authentication
	Helped Nicole with the README file

Full Name: Murad Almirov Student ID: GBC email:murad.almirov@georgebrown.ca	
Video URL	
GBLearn URL	
Tasks completed by member	UI design