**Project Title:** Learning Path Creator

**Completion date:**

Table of Contents

Contents

[Introduction 1](#_Toc149521438)

[Setup and Installation 1](#_Toc149521439)

[Application Architecture 1](#_Toc149521440)

[User Authentication and Authorization 1](#_Toc149521441)

[Database Interaction 1](#_Toc149521442)

[Server-Side Scripting (PHP) 2](#_Toc149521443)

[Responsive User Interface 2](#_Toc149521444)

[Error Handling and Validation 2](#_Toc149521445)

[Documentation and Code Quality 2](#_Toc149521446)

[Project-Specific Functionality 2](#_Toc149521447)

[Additional Notes 2](#_Toc149521448)

[Contact Information 2](#_Toc149521449)

Introduction

The purpose of Learning paths creator is to assist users in organizing and structuring learning content by creating paths that consist of various courses. Users can manage their own Learning Paths, view others’ Learning Paths. The app is featured in voting, cloning and sharing system, that encourages users to participate and involve in community learning.

Setup and Installation

To use our web application, user need to visit the link, registering if you don't have an account, or logging in if you already have one, Below is step-by-step instructions:

* Visit the Link: [Home (gblearn.com)](https://f3429112.gblearn.com/comp1230/assignments/project/)

Open your web browser and enter the provided link in the address bar.

* Check for Existing Account:

Look for an option on the website that allows you to check if you already have an account. This is typically labeled as "Login" .

* If You Have an Account:

Click on the "Login" link.

* Enter your username/email and password.

Click on the "Login" button.

* If You Don't Have an Account:

Look for an option to create a new account. This is labeled as "Register" .

Click on the "Register" link.

* Complete the Registration Form:

Fill out the registration form with the required information.

Typically, this includes providing necessary information like names, email address, and creating a password, etc

The websites might ask for additional details such as providing a profile photo, but this is optional.

* Submit the Registration Form:

After completing the registration form, click on the "Register" button.

* Verification (if required):

Ensure your email address is in the right format and valid.

* Login After Registration:

After successfully registering, go back to the login page.

Click on the "Login" link.

Enter your newly created email and password.

Click on the "Login" button.

* Explore the Website:

Once logged in, you should have access to the website's features and content.

Application Architecture  
The overall architecture of the application involves both the frontend and backend components, utilizing Bootstrap for appearance and cPanel for database and file management.

* *Frontend Architecture:*

1. HTML/CSS/JS:

The frontend is primarily built using HTML, CSS, and JavaScript.

Bootstrap is used as a front-end framework to enhance the appearance and responsiveness of the user interface.

* *Backend Architecture:*

1. PHP:

PHP is employed for server-side scripting and handling server-related tasks.

It interacts with the MySQL database to retrieve, insert, and update data.

1. MySQL Database:

cPanel is used for database management, providing an interface for creating and managing MySQL databases.

The application stores data, such as learning paths, courses, and URLs, in the MySQL database.

1. Database Interaction:

PHP scripts handle database interactions using the MySQLi or PDO extension.

Queries are executed to retrieve or modify data based on user actions.

1. Web Server and Hosting:

Web Server (e.g., Apache or Nginx):

The web server is responsible for serving the PHP files and handling HTTP requests.

It interprets PHP scripts and sends the corresponding HTML responses to the client.

1. cPanel:

cPanel serves as a control panel for managing various aspects of the hosting environment, including databases, files, domains, and security settings.

It provides a user-friendly interface for tasks such as creating databases, managing files, and configuring server settings.

***Summary:***

Frontend: HTML, CSS, JavaScript, Bootstrap.

Backend: PHP, MySQL.

Server: Apache or Nginx.

Database Management: cPanel.

User Authentication and Authorization

Handling user authentication and authorization is a critical aspect of a web applications to ensure secure access to user-specific data. Below is the outline of our basic approach to user authentication and authorization

User Authentication:

1. User Registration:

When a user registers, their password is securely hashed before storing it in the database.

Use a strong hashing algorithm bcrypt.

Store other user details (hashed password, email) in the database.

1. Login:

When a user logs in, verify their entered password against the hashed password in the database.

Use PHP's password\_verify function.

1. Session Management:

Use PHP sessions to manage user authentication state.

Store the user's ID, email, or other relevant information in the session upon successful login.

Include a few usernames(emails) and passwords:

…….

Database Interaction

1. Database Interaction with PDO:

Establish a PDO Connection

Perform SQL Queries

Schema:

A ER diagram prepresentation:

A screenshot of a computer screen

Description automatically generated

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)

Describe how server-side scripting is handled in the application and highlight any particular PHP functionalities used.

Responsive User Interface

Explain how responsiveness is achieved in the application’s user interface.

Error Handling and Validation

Detail the error handling and validation mechanisms in place throughout the application.

Documentation and Code Quality

Discuss the measures taken to ensure code quality and provide additional documentation if necessary.

Project-Specific Functionality (Innovative Feature)

List and describe all the project-specific functionalities implemented in the application.

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:  Student ID:  GBC email: | |
| Video URL |  |
| GBLearn URL |  |
| Tasks completed by member | Please provide a list of the tasks completed by this member. |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| Full Name:  Student ID:  GBC email: | |
| Video URL |  |
| GBLearn URL |  |
| Tasks completed by member | Please provide a list of the tasks completed by this member. |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| Full Name:  Student ID:  GBC email: | |
| Video URL |  |
| GBLearn URL |  |
| Tasks completed by member | Please provide a list of the tasks completed by this member. |
|  |
|  |
|  |
|  |