Project Title: Learning Path Creator

Completion date:

Table of Contents

Contents

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)

  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.

2. 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.

3. 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.

4. 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.

5. 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.

2. Login:

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

   Use PHP's password_verify function.

3. 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

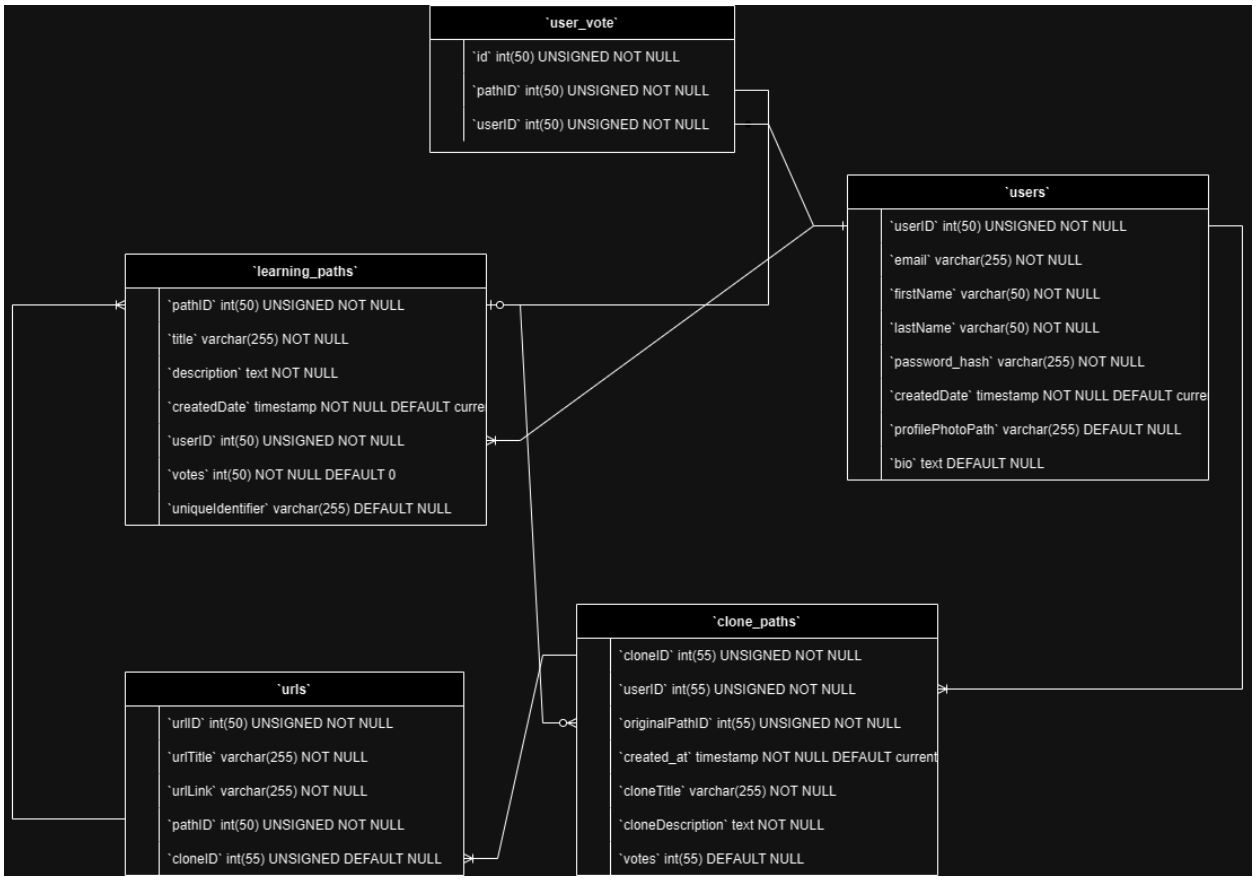
1 Database Interaction with PDO:

Establish a PDO Connection

Perform SQL Queries

Schema:

A ER diagram prepresentation:



Database Queries

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

$sql = "SELECT DISTINCT userID FROM users WHERE email = ?";	Selects unique users from the email table
$sql = "SELECT userID FROM user_vote WHERE userID = ?";	Checks if users exists in user_votes table
$sql = "SELECT DISTINCT userID FROM users WHERE email = ?";	Checks if unique User exists with a specic email in the users table
$sql = "INSERT INTO learning_paths (title, description, userID) VALUES (?, ?, ?)";	Insert values into the Learning_paths table

$sql = "SELECT title, description, userID, pathID FROM learning_paths WHERE pathID=?";	Grabs the information from a specific row of Learning path with a specific pathID
$sql = "SELECT title, description, userID, pathID FROM learning_paths WHERE userID=?";	Grabs the information from a specific row of Learning path with a specific userID
$sql = "SELECT urlID, urlTitle, urlLink FROM urls WHERE pathID = ?";	Get the url from a specific Path
$sql = "UPDATE learning_paths SET title=?, description=?, userID=? WHERE pathID=?";	Update learning path table
$sqlUrls = "UPDATE urls SET urlTitle=?, urlLink=? WHERE urlID=?";	Update urls table
$selectSql = "SELECT * FROM urls WHERE pathID = ?";	Grabs all Urls with a specific pathID
$insertSql = "INSERT INTO urls (pathID, urlTitle, urlLink) VALUES (?, ?, ?)";	Insert a new url into the url table
$sqlUrls = "DELETE FROM urls WHERE pathID = ?";	Delete a row from urls table with a specific pathID
$sql = "DELETE FROM learning_paths WHERE pathID =?";	Delete a row from learning_paths table with a specific pathID
$query = "SELECT votes FROM learning_paths WHERE pathID = :pathID";	Get the amount of votes for a learning path from the learning path table
$query = "SELECT * FROM user_vote WHERE pathID = :pathID AND userID = :userID";	Get all rows from user_vote table that has a specific pathID and userID
$query = "INSERT INTO user_vote (pathID, userID) VALUES (:pathID, :userID)";	Add a row to user_vote table that has a specific pathID and userID
$query = "DELETE FROM user_vote WHERE pathID = :pathID AND userID = :userID";	Remove a row from user_vote table that has a specific pathID and userID
$query = "UPDATE learning_paths SET votes = votes + :increment WHERE pathID = :pathID";	Increment the votes column in learning_paths with a specific pathID
$sql = "INSERT INTO urls (cloneID, pathID, urlTitle, urlLink) VALUES (?, ?, ?, ?)";	Enter a new row into the urls table
$sql = "INSERT INTO clone_paths (cloneTitle, cloneDescription, originalPathID, userID) VALUES (?, ?, ?, ?)";	Enter a new row into the clone_paths table
"SELECT pathID FROM learning_paths WHERE uniqueIdentifier = ?"	Get all pathIDs from rows that containe a uniqueIdentifier
"UPDATE learning_paths SET uniqueIdentifier = ? WHERE pathID = ?"	Update a the uniqueidentifer column of Leaning_paths in a row with a specific pathID
$sql = "SELECT DISTINCT * FROM clone_paths WHERE cloneID = ?";	Show all clone_paths with a specic cloneID
$sql = "SELECT password_hash FROM users WHERE email=? ";	Return the hashed password from the users table from a row with a specific email
$sql = "SELECT * FROM users WHERE email = ?";	Select all from the users table with a specic email row
$sql = "SELECT firstName, lastName FROM users WHERE email=? ";	Select first and last name from users table with a specific email row

$sql = "SELECT email FROM users WHERE email=? ";	Select email from the users table with a specific email
$sql = "INSERT INTO users (email, firstName, lastName, password_hash) VALUES(?,?,?,?)";	Add a new row into the users table

Server-side scripting (PHP)

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

function checkPathIDExist	Checks if PathID exists in the learning_path table
function checkUserExistsInUserVote	Checks if user exists in the user_vote table
function checkUserExists	Check if user exists in the user table
function setLearningPath	Set the information for a learning path
function updateSpecificLearningPath	Updates the information for a specific Learning path
function deleteSpecificLearningPath	Delate a specific learning path
function deleteSpecificLearningPath1	Delate a specific learning path and its children
function hasUserVoted	Checks if a specific User has voted on a specific learning path
function removeVote	Remove a vote from the larning path
function hasDownvoted	Checks if the user has Down voted
function updateVotes	Updates the votes for a specific Learning path
function setClonePath	Clone a path and copy the information into it
function insertUrls	Add information into the url table
function insertClonePath	Add information to the clone path
function generateUniqueIdentifier	Create a unique identifier
function getPathByUniqueIdentifier	Return a specific path by unique identifier
function storeIdentifier	Store the unique identifier in the database
function connect	Connect to the database
function emptyInput	See if the the row is empty
function createLearningPath	Creates a new learning path
function getUser	Get the User
function getUserName	Get the name of the User
function checkUser	Check if the user exists
function isValidEmail	Checks if it’s a valid email to create a user
function loginUser	Log in to a user
function setUser	Set the users information
function registerUser	Sign a user up
function myAutoLoader	Load the classes

Responsive User Interface

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

Responsive windows and contents in smal-medium-large sizes	
function displayLearningPaths	Show a specific learning path
function downvoteLearningPath	Downvote a specific learning path
function upvoteLearningPath	Upvote a specific learning path
function viewSpecificPath	View a specific path
function displayClonePathInfo	View Clone path information
function viewLearningPath	View specific learning learning path
function helloName	Greet the user
function getSpecificPath	Get a specific path
function getLearningPath	Get a specific Learning path
function getUrls	Get a specific URL
function getVote	Get the votes for a table
function handleVote	Logic for upvoteing, downvoting, and checking if voting is possible

Error Handling and Validation

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

function checkErrorCode	A large switch case stamen to give specific response to specific error

Documentation and Code Quality

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

The code is sectioned so each specific function is in a specific location as well as extensive comenting describing each function and what it does

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: Nhu Ly Student ID: 101429112 GBC email: HuynhYenNhu.Ly@georgebrown.ca	
Video URL	https://youtu.be/C_yvA4fujE8
GBLearn URL	https://f3429112.gblearn.com/comp1230/assignments/project/
Tasks completed by member	1. Create database schema
	2. Manage Learning Path Management Feature
	3. Manage User Authentication and Registration Feature
	4. Documented necessary parts
	5. Assisting team member in finishing Cloning, Voting and Sharing Systems
	6. Create User Interface

Full Name: Tomer Meir Chi Edelman Student ID: 101400506 GBC email: tomer.edelman@georgebrown.ca	
Video URL	https://youtu.be/-V8A7ens3mQ
GBLearn URL	https://f3400506.gblearn.com/
Tasks completed by member	Please provide a list of the tasks completed by this member.
	Upvote and Downvote system (non implentation)
	share learning path via Unique links (non implemntation)
	Clone and personilze learning path (non implmentation)

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.