

Project Title: Learning Path Creator

Completion date:

Table of Contents

Contents

Introduction.....	1
Setup and Installation.....	1
Application Architecture.....	1
User Authentication and Authorization.....	2
Database Interaction	2
Server-Side Scripting (PHP)	4
Responsive User Interface.....	4
Error Handling and Validation.....	6
Documentation and Code Quality	6
Project-Specific Functionality.....	6
Additional Notes.....	7
Contact Information	Error! Bookmark not defined.

Introduction

The Learning Path Creator is a web application designed to facilitate the creation, sharing, and exploration of learning paths. Learning paths are structured collections of resources and information that guide users through a particular topic or skill. This application empowers users to curate their learning content and share valuable paths with others in the community.

Setup and Installation

Clone the Repository:

git clone <https://github.com/ButterySpirit/learningpathcreator.git>

Create a MySQL database. Import the database.sql script.

Install PHP and configure server variables in PHP files.

Run composer install in the project root.

Adjust session settings in php.ini

Set up Apache or Nginx, with the document root in the public directory.

Open your browser and go to the app's URL.

Create an account or log in.

Begin creating paths and engaging with the community.

Application Architecture

The Learning Path Creator follows a web-based architecture, incorporating HTML, CSS, and JavaScript for the frontend. Bootstrap enhances design aesthetics, while jQuery handles asynchronous operations.

On the backend, PHP manages server-side scripting, facilitating user requests, database interactions, and dynamic content rendering. The MySQL database stores user data, learning paths, and associated information.

User authentication relies on PHP sessions, while authorization controls access based on user roles. The server environment includes Apache or Nginx, PHP, and MySQL.

The application employs additional libraries such as Bootstrap, jQuery, and Composer for frontend components, interactions, and PHP package management.

This architecture ensures scalability, maintainability, and responsiveness, promoting ease of development and future enhancements.

User Authentication and Authorization

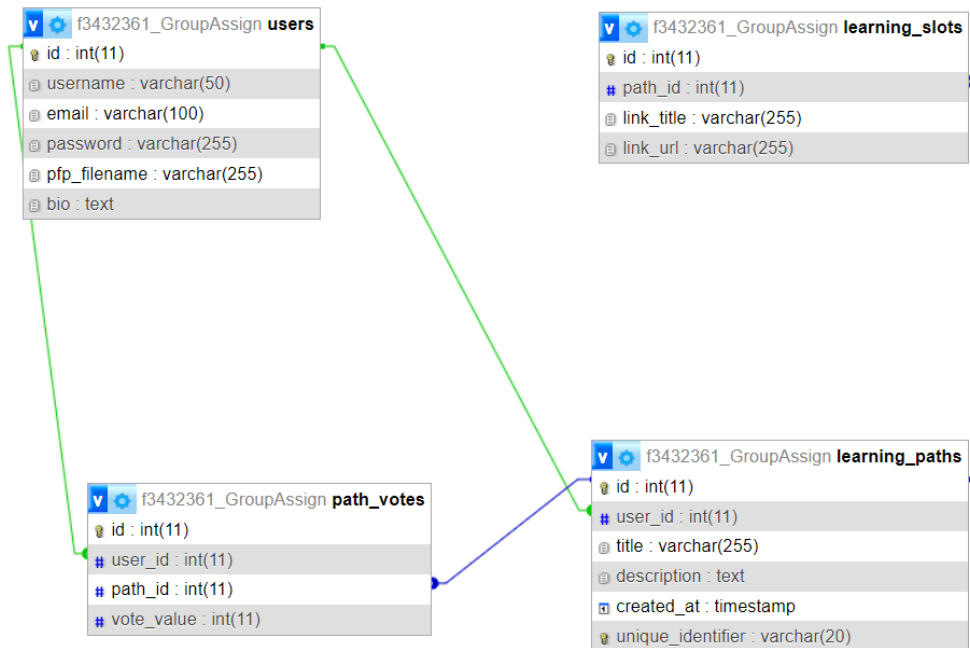
Users are required to log in using valid credentials through a dedicated login page. PHP sessions manage user authentication, providing a secure and seamless experience.

Username: a, Password:a , Username:b ,Password:b

Username:c,Password:c,Username:d,Password:d

Database Interaction

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



The Learning Path Creator interacts with the database using PHP's MySQLi extension. This choice facilitates secure and efficient communication with the MySQL database.

Database Connection:

Upon initiation, the application establishes a connection to the MySQL database hosted at "209.172.60.196." The connection credentials include the username "f3432361_User" and the associated password.

Prepared Statements:

To enhance security and prevent SQL injection attacks, the application extensively employs prepared statements. These statements are parameterized, ensuring that user inputs are treated as data rather than executable code.

Database Queries

Path Retrieval:

Purpose: Retrieve learning path details for display.

Query: `SELECT * FROM learning_paths WHERE id = ?`

Learning Slot Retrieval:

Purpose: Retrieve learning slots associated with a specific learning path.

Query:

Vote Count Retrieval:

Purpose: Calculate the total upvotes and downvotes for a learning path.

Query: `SELECT SUM(vote_value) AS total_votes FROM path_votes WHERE path_id = ?`

Search Paths:

Purpose: Search for learning paths based on title or description.

Query: `SELECT * FROM learning_paths WHERE title LIKE ? OR description LIKE ?`

User Authentication:

Purpose: Authenticate a user during the login process.

Query: `SELECT * FROM users WHERE username = ? AND password = ?`

User Registration:

Purpose: Insert a new user into the database during the registration process.

Query: `INSERT INTO users (username, password, email) VALUES (?, ?, ?)`

Create Learning Path:

Purpose: Insert a new learning path into the database.

Query: `INSERT INTO learning_paths (title, description, user_id) VALUES (?, ?, ?)`

Create Learning Slot:

Purpose: Insert a new learning slot into the database.

Query: `INSERT INTO learning_slots (path_id, link_title, link_url) VALUES (?, ?, ?)`

Vote Path:

Purpose: Record a user's vote (upvote or downvote) for a learning path.

Query: `INSERT INTO path_votes (path_id, user_id, vote_value) VALUES (?, ?, ?) ON DUPLICATE KEY UPDATE vote_value = VALUES(vote_value)`

Server-side scripting (PHP)

Server-side scripting in the Learning Path Creator is predominantly handled by PHP, providing dynamic content generation and server communication. The application utilizes various PHP functionalities to achieve its core features. Below are key aspects of server-side scripting:

Session Management: PHP's `session_start()` is employed for session management, enabling user authentication and tracking user-specific data throughout their interaction with the application. Sessions are crucial for maintaining user login states and personalized experiences.

Database Interaction: PHP is used to interact with the MySQL database through the MySQLi (MySQL Improved) extension. Prepared statements are extensively used to enhance security and prevent SQL injection attacks.

User Authentication: The application validates user credentials during login using PHP. The submitted username and password are checked against the database records, allowing or denying access accordingly.

User Registration: PHP facilitates the insertion of new user records into the database during the registration process. User-provided information, such as username, password, and email, is securely stored.

Learning Path Creation: New learning paths and associated learning slots are created using PHP. The server-side script processes user inputs, such as path title, description, and slot details, and inserts this information into the database.

Vote Handling: PHP functions handle the recording of user votes (upvotes or downvotes) for learning paths. The application uses AJAX to submit votes asynchronously without requiring a page refresh.

Error Handling: PHP's error handling mechanisms, such as `die()` and custom error pages, are implemented to gracefully handle errors during database connections, queries, and other critical processes.

AJAX Integration: PHP supports AJAX functionality for asynchronous communication with the server, particularly in features like voting and path sharing. The server-side script handles AJAX requests and responds with appropriate data.

The integration of these PHP functionalities ensures the robustness and dynamic nature of the Learning Path Creator, providing users with a seamless and interactive experience.

Responsive User Interface

Bootstrap Framework: Utilizes Bootstrap for a responsive grid system and pre-designed components.

Grid System: Organizes content using Bootstrap's 12-column grid system with responsive classes like `col-sm` and `col-md`.

Media Queries: Custom CSS media queries enhance responsiveness by adjusting styles based on screen width.

Flexible Images: Employs Bootstrap's `img-fluid` class to make images scale appropriately within containers.

Mobile-First Approach: Prioritizes mobile design, enhancing styles for larger screens progressively.

Responsive Navigation Bar: Adapts the navigation bar for smaller screens using Bootstrap's navbar-toggler class.

Fluid Containers: Utilizes Bootstrap's fluid containers for content that adjusts to various screen sizes.

Error Handling and Validation

Form Validation: Validates user inputs using HTML5 attributes and JavaScript for client-side checks. PHP ensures server-side validation, guaranteeing data integrity before processing.

Clear Messages: Presents user-friendly error messages for easy issue resolution. Clearly distinguishes between validation errors and critical system issues.

Database Constraints: Enforces database constraints to thwart invalid data at the source. Try-Catch Blocks: Utilizes try-catch blocks in PHP to handle exceptions gracefully. Captures errors, logs details, and prevents application crashes.

HTTP Status Codes: Employs appropriate HTTP status codes for transparent communication between client and server.

Logging: Implements logging mechanisms for debugging and continuous improvement.

User Feedback: Provides non-technical error messages, ensuring a positive user experience.

Documentation and Code Quality

The Learning Path Creator prioritizes code quality through meticulous documentation and adherence to industry best practices. Extensive inline comments clarify intricate sections, while project documentation provides insights into structure, functions, and variable meanings. Following PHP coding standards and a consistent naming convention enhances code readability and maintainability. Employing Git for version control facilitates efficient tracking of changes and collaborative development. Regular code reviews encourage adherence to standards and knowledge sharing. Thorough unit testing, continuous integration, and periodic security audits ensure robustness. Scalability considerations, code optimizations, and a separate documentation repository collectively contribute to a high-quality, secure, and maintainable codebase.

Project-Specific Functionality (Innovative Feature)

An innovative feature of the Learning Path Creator is the dynamic Explore page accessible from the dashboard. This page provides users with a comprehensive overview of learning paths across the entire platform, intelligently ordered by popularity. The ordering is determined by a voting system, where users can upvote or downvote learning paths based on their quality and relevance. The Explore page thus acts as a centralized hub, fostering collaboration and knowledge-sharing by highlighting the most acclaimed learning paths. This feature enhances user engagement, encourages exploration beyond individual paths, and promotes a sense of

community-driven learning. The implementation is seamlessly integrated into the dashboard, providing users with a curated selection of learning paths that align with community preferences.

Additional Notes

Have fun!

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: Nikola Varicak Student ID:101432361 GBC email:101432361@georgebrown.ca	
Video URL	https://youtu.be/oq2sa_faqKg
GBLearn URL	https://f3432361.gblearn.com/COMP1230/
Tasks completed by member	User Registration and Authentication → Connection to DB
	Learning Path Management
	Sharing and Cloning
	Search Functionality

Full Name: Ali Mouasvi Roozbahani Student ID: 101321818 GBC email:101321818@georgebrown.ca	
Video URL	https://www.youtube.com/watch?v=2CggglqmckU
GBLearn URL	https://f3321818.gblearn.com/comp1230/
Tasks completed by member	User Reg and Authentication
	Voting System
	Profile Management
	Innovative Feature
	Bootstrap Styling

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.
