

Michaela Zammit



GAMING SOCIAL MEDIA

{website on github}

Database Set Up

In phpMyAdmin, I log in and pick the database I'm interested in. Then, I look at the tables and choose one. I check out its structure and data. If I need to do something specific, I can write commands in the "SQL" tab. This helps me understand and work with the database better.

Techniques to manipulate Data

In my web project, I've set up ``functions.php`` and ``db.php`` to handle all things related to my database. ``db.php`` is like the backbone of my project; it connects everything to my database by holding details like the host, username, password, and database name. It's the first thing my project leans on when it needs to interact with data.

Then there's ``functions.php``, which is where I keep all my custom-made tools for working with the database. I've written functions in there for every job imaginable: inserting new data, updating existing records, fetching information, you name it. It's like my toolkit for tinkering with the database.

Together, these files make my project organized and efficient. Whenever I need to manipulate data, I just dive into ``functions.php``, pick the right tool, and get to work. It's my way of keeping things tidy and making sure my project runs smoothly.

To mimic a live server on my own device, I chose XAMPP. With it, I started up Apache and MySQL – they're like the backbone, handling web pages and databases. After a bit of setup tweaking, I used phpMyAdmin to manage databases for my projects.

Then, I organized my projects neatly in XAMPP's folders, ready for testing. I accessed them in my browser using `localhost`, just like they'd be online. It was a safe space to tweak and fix things before they went live. XAMPP made it all easy and fuss-free.

Techniques

In my web projects, I've found a trio that works wonders: PHP, CSS, and HTML. HTML sets up the skeleton of my pages, CSS styles them to look good, and PHP adds the smarts, making things dynamic.

HTML is where I sketch out the layout of my site, like drawing on a canvas. CSS comes in to paint the colours and shapes, making everything look neat and tidy.

Then, PHP steps in to make things move. It's like the engine behind the scenes, handling things like form submissions, interacting with databases, and making content change based on user actions.

Together, they make building dynamic web apps feel like putting together puzzle pieces – each part plays its role, and the result is a smooth, interactive experience for users.

Test Cases

Test	Input	Process	Output	Actual Output	Pass/Fail
1	Follow	Press follow	Follow person	Person followed	Pass
2	Like	Liking Video	Video Liked	Liked Video	Pass
3	Comment	Write comment	Comment posted	Comment Posted	Pass
4	Share	Sharing Video	Video Shared	Video Shared	Pass
5	Post	Upload Video	Video uploaded	Video uploaded	Pass
		Add title			
		Add description			
6	Add Challenge	Upload Video	Challanged added	Challanged added	Pass
		Add title			
7	Post Challenge	Upload Video	Challanged posted	Challanged posted	Pass
		Add title			
		Add description			
8	Message	Click on the person	Message sent	Message sent	Pass
		Send message		Message received	
9	Profile Picture	Add picture	Picture Added	Picture Added	Pass
				Picture Changed	