Html css javascript php mysql tutorial pdf

I'm not robot

reCAPTCHA

Continue

No matter where and by whom you take your lessons, you should always do all the hard work yourself. You must spend time and effort learning, understanding, memorizing and practicing new knowledge. That's not a short circuit here. But a good course can show the best route and help you avoid traps and detours. You can achieve maximum productivity from your time and efforts. Based on user feedback, technology development, and our new thoughts and inspirations, we will continue to add new content and improve existing ones. We will cover the basics as well as advanced applications. You'll learn smart and creative technology combinations as well. Web development has been a popular site for years. The internet has accumulated tons of free teaching videos and materials. Even if you're stuck with questions you can't solve, just post them in stackoverflow, you'll have your answers within minutes. With all the free teaching materials and teachers, what is the point of creating this paid online course? Depends on what you really want. If you just want to have a rough idea of how web applications work and create some basic programs, then there's really no need for you to pay for any courses. Free information on the internet is more than enough to meet your requirements. If you want to be able to turn your design and idea into programs that are safe and effective enough to be implemented in real life and solve all kinds of challenge, then this lesson is perfect for you. We believe that a good online course should also be a navigator, not just explaining coding concepts. It should show you where the real destination is and most importantly, take you there in the most efficient way. Why are we emphasizing the real destination? First of all, you need to know a program that can work is very different from a program that can survive in real life. This is like playing a racing game on your phone is definitely not the same as driving a real car on a real track. For example, in our PHP course, we teach you how to create an upload program. We've seen books and lessons that only teach how to upload files. If you run this program in real life, it won't last a day. It'll do more damage than good. For a real-life upload program to work, you must check the file type and size before letting any file be uploaded. You must also protect your program from sending the file repeatedly, maintaining the refresh of the upload page. Sent files should be renamed and saved correctly. Most importantly, this process must be automatic. Loaded files must also be recoverable, so information about loaded files must also be automatically recorded in the database. If one step is missing, the upload program will not survive the real-life environment. That is why we are emphasising the real destination. We want to show you the whole picture, not just part of it. Knowing the whole picture is what separates a trusted professional programmer from a coding amateur. Obviously the road to real destination is quite large. This is why a good course should be a navigator. You have to show the students the best route. When designing the content of our course, we recognize and most importantly, take into account two very important facts: Everyone forgets and you have other things going on in your life and can only partially devote to your study. This means that your study will be discontinued continuously and when you continue your study, you may find what you have learned has become rusty. As a result, all your new knowledge is built on a crumbling institution. At the end of your study, you will find that despite the fact that you have spent time and money, but you simply cannot make sufficient progress. You thought you've learned everything, but just can't put them together can create a functional program. How are we going to solve this problem? We divide the entire content of the course into small sections. At the end of each section, there will be a review session that summarizes everything you've learned from that session. When you create APPs, we'll give a review session on basic knowledge as well. In addition to review sessions, we will also help you control the pace of your study. You will often hear us tell you that if you can't answer this quiz, don't go ahead and review the previous tutorials. The biggest card up our sleeve is our content board. You may think the upload program course starts here. But in fact our upload course has started long before that. We have been starting to lay the foundation for you since we taught you how to create a file type detector. We divide a complex program into items and we will only add one item at a time. Most importantly, every time we add a new item, we will look again at the previous elements. In the end, despite the program we create being long and complicated, you will be able to master everything. EXPLANATORY CODE DOWNLOADClick here to download the source code, I have released with the MIT license, so feel free to build on it or use it in your own work. FAST NOTES Take and unzip to a folder. Create a virtual database and import 1-products.sql.Update 2-products.php and change the database settings for your own. Start 3-html-page.php in your web browser and follow along. If you find an error, feel free to comment below. I try to answer questions too, but it's one person against the whole world... If you need answers urgently, please check out my list of websites to get help with programming. DIVIDE AND it is common for beginners to be paralyzed by too many things to deal with and do not know where to start. So the first step in any project is to break things down and define a step-by-step plan. So in this example of creating a simple product page:Goal & Overview: As captain obvious as this may be, we need to create a page to display the products from the database. Server-side database (MySQL): To keep things simple, the database will record only the name and description of the product. PHP server-side: Create a receiving product information from the database. Client-Side HTML & CSS: Use the PHP script to create an HTML page to display products, CSS for style. Client-side Javascript: Finally, use javascript to add dynamic elements to the HTML page. For example, what happens when the user clicks a product. STEP 1. MYSQL DATABASETHE SQLCREATE TABLE 'products' ('product_id' int(11) NOT NULL, 'product_name' varchar(128) NOT NULL, 'product_description' text) ENGINE=InnoDB DEFAULT CHARSET=latin1; INTERESTED IN products (product_id, product_name, product_description) VALUES (1, Sculpture Of Fortune, A sweaty blow is produced along the divider here.), (2, Guardian Without Duty, Does a painting migrate within an excessively paranoid?), (3, Enemies Without Hope, A Cured Parameter Fears Behind the Phenomenon.) (4, Lords Of The Void, The Diary Scores Around The Generalized Lie) (5, Doctors and Aliens, The Diary Scores Around The Generalized Lie.) (6, 'Ironsmiths and criminals') AMEND THE 'products' TABLE FIRST KEY ADD ('product_id'), ADD 'product_name' KEYS ('product_name'); AMENDMENT OF TABLE 'products' AMENDMENT OF 'product_id' int(11) NO NULL AUTO_INCREMENT, AUTO_INCREMENT=7; BANK STRUCTUREFieldDescriptionproduct_idProduct ID. The primary key, the name of the auto-increment.product_nameThe. The index to search performance.product_descriptionThe product description. MYSQL DATABASE NOTES & EXPLANATIONPersonally, I will always start with the database, simply because it is the foundation of almost every project. It is important to get the required fields right from the beginning, or changing the database structure later can be very painful later. STEP 2. PHPTHE SCRIPT <?php (a)= connect= to= database= != change= settings= to= your= own= != define('db_host',= 'localhost');= define('db_name',= 'test');= define('db_charset',= 'utf8');= define('db_user',= 'root');= define('db_password',= '');= try= {= $pdo=new pdo(= mysql:host= . DB_HOST . ;charset= . Db_charset. ;dbname= . DB_NAME, DB_USER, DB_PASSWORD, [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]); } catch (Exception $ex) { die($ex->getMessage()); } // (B) GET PRODUCTS $stmt = $pdo->prepare(select= *= from= 'products'$stmt); $products = $stmt->fetchAll(PDO::FETCH_NAMED); PHP NOTES & EXPLANATIONFAP is the next piece of the puzzle and the foundation, the above is an overly simplified example of how we can use php to bring data from the database ... It's really kind of bad, but remember that in real works, this would have been done in a library or put in a class, object-oriented style. STEP 3. HTML THE SCRIPT html><html><head><title> Σελίδα επιδείξης προϊόντων </title><!-- (A) CSS & JS --> <link href=4-style.css rel=stylesheet> <script src=5-script.js></script></head><body><!-- (B) BOOKS LIST --><h1>TA BIBAIA ΜΑΣ</h1><div id=our-books><?php require= 2-products.php;= foreach= ($products= as= $p)= {=?><div class=book-wrap class=book-wrap><div class=book-wrap ></div></body></html> ></div></body></html> <div class=book-title><?=$p['product_name']?> ></div><div class=book-desc><?=$p['product_description']?> ></div><?php }=?>HTML NOTES & EXPLORE what's going on here!? Why is the PHP file extension instead of HTML? That's right - We use PHP alongside HTML here. You don't have to get confused. Do you remember the script we wrote earlier to get product data from the database? We're just using it on this page to create the HTML, that's all. STEP 4. THE CSSTHE SCRIPT/* (A) WHOLE PAGE */ html, body { font-family: arial, sans-serif; }* (B) BOOKS */ #our-books { max-width: 1200px; margin: 0 auto; display: grid; grid-template-columns: auto; grid-gap: 10px; } #our-books .book-wrap { background: #f1f1f1 replenishment: 15px; border: 1px solid #808080; } #our books .book-wrap:hover { cursor: pointer; background: #fdffe6; } #our-books .book-title { font-weight: bold; font-size: 1.2em; color: #333; } #our-books .book-desc { margin-top: 10px, color: #848484; CSS NOTES & EXPLANATION Yes that's a lot you're actually doing here. CSS is the place that deals with cosmetics and styles in HTML. STEP 5. JAVASCRIPTTHE SCRIPTwindow.addEventListener(load, function(){ for (leave document.getElementsByClassName(book-wrap)) { book.addEventListener(click, function(){ var id = this.dataset.id, name = this.getElementsByClassName(book-title)[0].innerHTML, desc = this.getElementsByClassName(book-desc)[0].innerHTML; alert('You have selected - ID: ${id}, TITLE: ${name} DESC: ${desc}'); }); }); }); JAVASCRIPT NOTES & EXPLANATION Good right, another confession to make, this is too simplified. On a normal product page, Javascript would have been used to send an add-to-cart process to the server when you click an item. But you can get the drift, Javascript is used to make everything dynamic - Add to basket, remove from basket, submit forms, checks, load more content, change themes, etc ... RESULTYep, this is not the most beautiful looking award-winning website on the planet. But we need to do well enough to demonstrate how we can use all PHP, MySQL, HTML, CSS, and Javascript together. Software Development Life (SDLC)Congratulations, what you've spent in the last 15 minutes is called software development lifecycle (SDLC). Requirement analysis – What is required? What does the customer need? Design - Divide-and-conquer. What's to be done? Design - Database structure, software architecture. Development – Actual encoding. Testing & Deployment – Test the code, put it live on the Internet.Some people might have different steps for the SDLC, but that is the general process. Also, note that the SDLC does not stop after a single cycle – It can be our projects later and decide that things need to change. The whole SDLC cycle begins again, and never really ends. FRONT-END, BACK-END, FULL-STACKAs beginner, you don't have to worry too much actually. In the real world huge projects, there are usually many developers working on the project, any handling of a different item: Project Manager (Full-Stack) - This guy is your boss. It deals with broad overview, processes, system-wide structure, and even support for other platforms, such as mobile apps. Lead / Senior Web Developers (Full-Stack) - Does everything. HTML, CSS, Javascript, PHP, MySQL, and everything else. AKA Superman.Junior Web Developers (Front End) - Creates Web pages. HTML, CSS, Javascript.Junior Web Developers (Back End) - Makes PHP, MySQL. More about the administrative table and server-side libraries. So yes, it's only higher up the ladder that we really have to deal with everything and beyond. LINKS & REPORTS How HTML, CSS, PHP, Javascript, MySQL Work Together (Click to Enlarge) Thank you for reading, and we have reached the end of this guide. I hope it has helped you understand better - HTML, CSS, Javascript, PHP, MySQL, they are all different elements, but they all work together to create a single website or system. It can be overwhelming at first to deal with many things at once, so take it step by step, and the rest will be just like learning how to cycle. Once you catch the stuff and flow, it will be much easier. If you want to share something with this guide, feel free to comment below. Good luck and happy coding! Encoding!