DEVELOPING E-COMMERCE Web Apps with PHP & MySQL



Web Browser

Advanced Web Technologies

□ Aims of Advanced Web Technologies:

- ✓ Server platform independence
- ✓ Client Technology Independence
- ✓ Formalizing the communication between the different components:
 - \circ Web Client \rightarrow Web Server,
 - Application Server \rightarrow DB Server
- ✓ Deployment of a distributed model of implementing software and applications

All the Web Technologies are based on protocols and various Internet Services.

Prerequisite:



All the Web Technologies are based on protocols and various Internet Services.

Prerequisite:



All the Web Technologies are based on protocols and various Internet Services.

Prerequisite:



All the Web Technologies are based on protocols and various Internet Services.

Prerequisite:



All the Web Technologies are based on protocols and various Internet Services.

Prerequisite:



All the Web Technologies are based on protocols and various Internet Services.

Prerequisite:









	OSI	TCP/IP
7	Application	Applications (FTP, SMTP, HTTP, etc.)
6	Presentation	
5	Session	
4	Transport	TCP (host-to-host)
3	Network	IP
2	Data link	Network access (usually Ethernet)
1	Physical	





All the Web Technologies are based on protocols and various *Internet Services*.

✓ <u>Software-as-a-Service / SaaS</u>

The application is executed on the Web Server and Application Server

- Thin Client
- In other words:
 - 1. Accessing and managing software through networking processes
 - 2. Various actions are managed centrally (software update and upgrades), allowing clients to access apps through the Web.
 - 3. Multi-Tenant Architecture.

Client – Server Model

However, in practice we have the so called... *thin client - fat server* model



Client – Server Model

Web Client



Client – Server Model

Web Client

 Implement various protocols (e.g. http) and scripting languages at the side of Client (Client Side Scripting).



Client – Server Model

Web Client

- Implement various protocols (e.g. http) and scripting languages at the side of Client (Client Side Scripting),
- ✓ Rendering the layout.



Client – Server Model

Web Client

- Implement various protocols (e.g. http) and scripting languages at the side of Client (Client Side Scripting),
- \checkmark Rendering the layout,
- Client Scripting (manipulating data with JavaScript).



Client – Server Model

Web Client

- Implement various protocols (e.g. http) and scripting languages at the side of Client (Client Side Scripting),
- \checkmark Rendering the layout,
- Client Scripting (manipulating data with JavaScript),
- ✓ Web Clients \rightarrow Web Browsers :

MS IExplorer, Mozilla, Google Chrome, Netscape, Opera, etc.



Client – Server Model

Web Client

- Implement various protocols (e.g. http) and scripting languages at the side of Client (Client Side Scripting),
- \checkmark Rendering the layout,
- Client Scripting (manipulating data with JavaScript),
- ✓ Web Clients \rightarrow Web Browsers :

MS IExplorer, Mozilla, Google Chrome, Netscape, Opera, etc.



Client – Server Model

Web Client

- Implement various protocols (e.g. http) and scripting languages at the side of Client (Client Side Scripting),
- \checkmark Rendering the layout,
- Client Scripting (manipulating data with JavaScript),
- ✓ Web Clients \rightarrow Web Browsers :

MS IExplorer, Mozilla, Google Chrome, Netscape, Opera, etc.



Client – Server Model



Client – Server Model

Web Server

✓ Host (serve) dynamic or static pages.



Client – Server Model

- ✓ Host (serve) dynamic or static pages.
- ✓ Manage Various Processes:
 - 1. Sessions,
 - 2. Interpretation,
 - 3. Securing Communications,
 - 4. Server Side Scripting Interaction.



Client – Server Model

- ✓ Host (serve) dynamic or static pages.
- ✓ Manage Various Processes:
 - 1. Sessions,
 - 2. Interpretation,
 - 3. Securing Communications,
 - 4. Server Side Scripting Interaction.
 - Web Servers: MS IIS, Apache.



Client – Server Model

- ✓ Host (serve) dynamic or static pages.
- ✓ Manage Various Processes:
 - 1. Sessions,
 - 2. Interpretation,
 - 3. Securing Communications,
 - 4. Server Side Scripting Interaction.
 - Web Servers: MS IIS, Apache.





- ✓ Basic Type of Web Apps
 - o Client-Server is a Many-to-Many Relation
 - o A Web Server may serves simultaneously many Web Clients
 - o A Web Client may be connected to many Web Servers

□ Anatomy of a Client – Server Model



Web Browser









CLIENT SERVER PRELIMINARIES

Model Architecture

✓ The client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called Servers, and service requesters, called Clients.


Model Architecture

- ✓ The client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called Servers, and service requesters, called Clients.
- ✓ Often Clients and Servers communicate over a computer network on separate hardware, but both Client and Server may reside in the same system.



Model Architecture

- ✓ The client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called Servers, and service requesters, called Clients.
- ✓ Often Clients and Servers communicate over a computer network on separate hardware, but both Client and Server may reside in the same system.
- ✓ A Server host runs one or more Server programs which share their resources with clients.



Model Architecture

- ✓ The client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called Servers, and service requesters, called Clients.
- ✓ Often Clients and Servers communicate over a computer network on separate hardware, but both Client and Server may reside in the same system.
- ✓ A Server host runs one or more Server programs which share their resources with clients.
- ✓ A Client does not share any of its resources, but requests a Server's content or service function.
 - Clients initiate communication sessions with Servers which await incoming Requests ...



\Box Client Server Role (1/2)

 ✓ The Client-Server model describes the relationship of cooperating programs in an application.



- ✓ The Client-Server model describes the relationship of cooperating programs in an application.
- The Server component provides a function or service to one or many Clients,



- ✓ The Client-Server model describes the relationship of cooperating programs in an application.
- The Server component provides a function or service to one or many Clients, which initiate requests for such services.



- ✓ The Client-Server model describes the relationship of cooperating programs in an application.
- ✓ The Server component provides a function or service to one or many Clients, which initiate requests for such services.
- ✓ Usually, Servers are classified by the services they provide.



- ✓ The Client-Server model describes the relationship of cooperating programs in an application.
- ✓ The Server component provides a function or service to one or many Clients, which initiate requests for such services.
- ✓ Usually, Servers are classified by the services they provide.
- ✓ A shared resource may be any of the Server computer's software and electronic components.



- ✓ The Client-Server model describes the relationship of cooperating programs in an application.
- The Server component provides a function or service to one or many Clients, which initiate requests for such services.
- ✓ Usually, Servers are classified by the services they provide.
- ✓ A shared resource may be any of the Server computer's software and electronic components.
- ✓ What is a <u>SERVICE</u>: The sharing of resources of a Server.



\Box Client Server Role (1/2)

- ✓ The Client-Server model describes the relationship of cooperating programs in an application.
- ✓ The Server component provides a function or service to one or many Clients, which initiate requests for such services.
- ✓ Usually, Servers are classified by the services they provide.
- ✓ A shared resource may be any of the Server computer's software and electronic components.
- ✓ What is a <u>SERVICE</u>: The sharing of resources of a Server.
 - Services are an abstraction of computer resources.
 - A client does not have to be concerned with how the server performs while fulfilling the request and delivering the response.
 - The client only has to understand the response based on the protocol (i.e. the content and the formatting of the data for the requested service)



https://en.wikipedia.org/wiki/Client%E2%80%93server_model

\Box Client Server Role (2/2)

✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.



- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.



- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.
 - Clients and Servers exchange messages in a request-response messaging pattern.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

• The Client sends a request, and the Server returns a response (inter-process communication).



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- The Client sends a request, and the Server returns a response (inter-process communication).
- To communicate, the computers must have a common language, and they must follow rules so that both the Client and the Server know what to expect (communications protocol).



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- The Client sends a request, and the Server returns a response (inter-process communication).
- To communicate, the computers must have a common language, and they must follow rules so that both the Client and the Server know what to expect (communications protocol).
- All Client-server Protocols operate in the Application Layer. https://en.wikipedia.org/wiki/Client%E2%80%93server model



- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.
 - Clients and Servers exchange messages in a request-response messaging pattern.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

• The application layer **Protocol** defines the basic patterns of the dialogue.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between <u>Servers</u>, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- The application layer **Protocol** defines the basic patterns of the dialogue.
- To formalize the data exchange even further, the Server may implement an application programming interface (API).



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between <u>Servers</u>, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- The application layer **Protocol** defines the basic patterns of the dialogue.
- To formalize the data exchange even further, the Server may implement an application programming interface (API).





- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.
 - Clients and Servers exchange messages in a request-response messaging pattern.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

• By restricting communication to a specific content format, it facilitates parsing.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between <u>Servers</u>, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- By restricting communication to a specific content format, it facilitates parsing.
- By abstracting access, it facilitates Cross-Platform data exchange.



https://en.wikipedia.org/wiki/Client%E2%80%93server_model

\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- By restricting communication to a specific content format, it facilitates parsing.
- By abstracting access, it facilitates Cross-Platform data exchange.
- A Server may receive requests from many distinct Clients in a short period of time.





- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.
 - Clients and Servers exchange messages in a request-response messaging pattern.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

• A computer (Server) can only perform a limited number of tasks at any moment!!!



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- A computer (Server) can only perform a limited number of tasks at any moment!!!
 - ... relies on a scheduling system to prioritize incoming requests from Clients for accommodation.



\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between Servers, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

- A computer (Server) can only perform a limited number of tasks at any moment!!!
 - ... relies on a scheduling system to prioritize incoming requests from Clients for accommodation.
- To prevent abuse and maximize availability, Server software may limit the availability to Clients.



https://en.wikipedia.org/wiki/Client%E2%80%93server_model

\Box Client Server Role (2/2)

- ✓ Whether a computer is a Client, a Server, or both, is determined by the nature of the application that requires the Service functions.
- ✓ What is <u>Server-to-Server communication</u>: communication between <u>Servers</u>, such as to synchronize data.

Clients and Servers exchange messages in a request-response messaging pattern.

The Denial of Service - DOS attacks are designed to exploit a Server's obligation to process requests by overloading it with excessive request rates.



Anatomy of a Client – Server Model Recall: All the Web Technologies are based on protocols !



Anatomy of a Client – Server Model Recall: All the Web Technologies are based on protocols !



Anatomy of a Client – Server Model Recall: All the Web Technologies are based on protocols !

✓ <u>HTTP Protocol</u>

Answer - Response Model

- o Answers start from the Web Client,
- Responses start from the Web Server,
- There may exist intermediate nodes...



U Web Technologies



□ Web Technologies

Application Servers

- They combine:
- o Web Servers,
- Abstract access to data and databases,
- o Clustering / Data Availability,
- Messaging buses.





□ Web Technologies

<u>Data Bases</u>

- They store: data, credentials, app. usage data, etc.
- Interact with application servers utilizing:
 - 1. SQL (ODBC, JDBC)
 - 2. Object-Oriented DB Access



□ Web Technologies

WEB Services

- Defined by W3C (Web Consortium),
- Formalization of the interaction between Web Apps,
- Software system for the compatibility of the interaction between different machines.






BASIS OF ADVANCED WEB TECHNOLOGIES

□ Web Technologies

WEB Services

- Defined by W3C (Web Consortium),
- Formalization of the interaction between Web Apps,
- Software system for the compatibility of the interaction between different machines.





The Universal Description, Discovery and Integration (UDDI) specifications define a registry service for Web services and for other electronic and nonelectronic services.

A UDDI registry service is a Web service that manages information about service providers, service implementations, and service metadata.

BASIS OF ADVANCED WEB TECHNOLOGIES

□ Web Technologies

WEB Services

- Defined by W3C (Web Consortium),
- Formalization of the interaction between Web Apps,
- Software system for the compatibility of the interaction between different machines.





The Web Services Description Language is an XML-based interface definition language that is used for describing the functionality offered by a web service.

The acronym is also used for any specific WSDL description of a web service, which provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns.

BASIS OF ADVANCED WEB TECHNOLOGIES

□ Web Technologies

WEB Services

- Defined by W3C (Web Consortium),
- Formalization of the interaction between Web Apps,
- Software system for the compatibility of the interaction between different machines.

Service Broker UDDI WSDL WSDL WSDL SOAP Service Requester Service Provider



SOAP (Simple Object Access Protocol) is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks.

Its purpose is to induce extensibility, neutrality and independence.

COMPONENT ARCHITECTURE...







□ Hyper Text Markup Language:



□ Hyper Text Markup Language:

✓ A document containing tags (markup),



□ Hyper Text Markup Language:

- ✓ A document containing tags (markup),
- ✓ Browser interprets the tags rendering the content,



□ Hyper Text Markup Language:

- ✓ A document containing tags (markup),
- ✓ Browser interprets the tags rendering the content,
- Easy to develop using tools ranging from simple text editors to more advanced web design tools.





□ Hyper Text Markup Language:

✓ A document containing tags (markup),

✓ Browser interprets the tags rendering the content,

✓ Easy to develop using tools ranging from simple text editors to more advanced web design tools.





Anatomy of a Hyper Text Markup Language Document: <html> <head> <title>Page Title!</title> </head> <body> This is my first page !!! </body> </html>



□ Anatomy of a Hyper Text Markup Language Document:

<html>
<head>
<title>Page Title!</title>
</head>
<body>
This is my first page !!!
</body>

</html>



□ Anatomy of a Hyper Text Markup Language Document:

<head>

<title>Page Title!</title>

</head>

• <u>head section</u>:

TAG	Description
title	Defines page's title
script	Defines scripting language
style	Defines CSS
•••	
meta	Defines metadata → SEO





Anatomy of a Hyper Text Markup Language Document: <html> <head> <title>Page Title!</title> </head> <body> This is my first page !!! </body> </html>



Anatomy of a Hyper Text Markup Language Document: <html> <head> <title>Page Title!</title> </head> <body> This is my first page !!! </body> </html>

86



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section</u>:

TAG	Description
bgcolor	Defines background's color
text	Defines text's color
link	Defines link's color
•••	
background	Defines background image





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section</u>:

TAG	Description
h(1-6)	Defines header style
р	Defines paragraph
u/o/d 1	Defines lists
•••	
form	Defines forms





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section</u>:

TAG	Description
table	Defines table
hr	Defines lines
a	Defines links
•••	
img	Defines image





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section</u>:

TAG	Description
b	Defines bold fonts
i	Defines italic fonts
u	Defines <u>underline</u> fonts
•••	
br	Defines line change





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

TAG	Description
form	Defines aform for user input
input	Defines an input control
textarea	Defines a multiline input control
label	Defines label for an <input/>





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

TAG	Description
option	Defines an option in a drop-down list
button	Defines a clickable button
select	Defines a drop-down list
optgroup	Defines a group of related options in a drop-down list



92



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	button
code	<input onclick="myfunction()" type="button" value="Send"/>
description	If it is a simple button and not a submit button, you need to associate JavaScript code with the onclick event to execute something.
appearance	Button



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

object	Text field	
code	<input name="mytext" type="text" value="default text"/>	
description	Other attributes are: type = "password" for a field to the hidden content, size = width of the text field, maxlength = max # characters.	
appearance	default text	



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	Text area	
code	<textarea name="textarea">content</textarea 	
description	Text field on several lines. The difference with the previous object is that the initial content is placed between the starting and ending tags, not in an attribute.	
appearance	content	

95



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

object	Password
code	<input name="pass" type="password" value="12345"/>
description	
appearance	[]



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	Hidden field
code	<input type="hidden" value="code"/>
description	Allows to add data to values sent by the form, that are defined by a script rather than entered by the user.
appearance	





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	Checkbox	
code	<input checked="" name="cb1" type="checkbox" value="true"/>	
description	The checked attribute allows you to check the box initially. An article is dedicated to the study of the use of check boxes.	
appearance		



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

object	Radio button	
code	<input name="radiox" type="radio" value="radio1"/>	
description		NIX.
appearance		



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

object	Radio group	
code	<input <br="" type="radio"/> name="id1"value="true" checked> Yes <input <br="" name="id1" type="radio"/> value="false"> No	
description	A radio button works as a checkbox. But a group of radio buttons allows an alternative: checking one unchecks others. So that the buttons are alternative, we must give the same name or id for all.	
appearance	💿 Yes 🔍 No	

100



101

□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	Menu
code	<pre><select name="select"> <option>cherry</option> <option>orange</option> <option>apple</option> </select></pre>
description	The select tag and option inner tags allow to build a menu that can take the form of a scrolling list or a static list.
appearance	cherry T



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	List	
code	<pre><select name="select2" size="3"> <option>cherry</option> <option>orange</option> <option>apple</option> </select></pre>	
description		
appearance	cherry orange apple	



102



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

object	File		
code	<input name="file" type="file"/>		
description	HTML has a file selection function with the file type "file" which combines a text field and a button. The local filename is assigned to the value attribute of the object. The content of the local file is added to form data and sent with them.		
appearance	Επιλογή αρχείου Δεν επιλέχθηκανένα αρχείο.		





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o <u>body section - forms</u>:

object	Image
code	<input name="image" src="xulfr.gif" type="image"/>
description	The src attribute indicates the URL of the image. The difference with the img tag is that it is taken into account in the form data passed to another page.
appearance	





105

□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

• <u>body section - forms</u>:

object	Submit	
code	<input type="submit" value="Submit"/>	
description	It send the form data to a script or another page. The value attribute defines the label to display on the button. When you press this button, the file defined by the action attribute of the form is loaded in place of the current page and the form data are given as parameters. We associate an onclick event to the submit buttons.	
appearance	Submit	

34



Anatomy of a Hyper Text Markup Language Document: <u>A complete form example</u>:

File Edit	Search View Encoding Language Settings Macro Run Plugins Window ? 🔄 🗞 😘 🕼 📣 🐇 瞻 🖺 💭 📽 📾 🎥 象 🤏 🖫 💁 🚍 🗐 🎼 🕼 🕼 🖉	/C:/Users/User/Desktop/test.html × +
test.htm		
1	S <html></html>	$(\leftarrow) \rightarrow C \land$
2		
3	<head></head>	
4		
5 6	⊇ <body></body>	button - Send
7	<pre>state="myform" id="myform" action="page.html" onSubmit="return fun()"></pre>	text - default text
8	<pre> button - <input onclick="myfunction()" type="button" value="Send"/></pre>	leaf defour text
9	<pre> text - <input name="mytext" type="text" value="default text"/></pre>	content
10	<pre> textarea - <textarea name="textarea">content</textarea></pre>	
11	<pre> password - <input name="pass" type="password" value="12345"/></pre>	
12	<pre> hidden - <input type="hidden" value="code"/></pre>	textarea
13	<pre> checkbox - <input checked="" name="cb1" type="checkbox" value="true"/></pre>	nassword -
14	<pre> radio - <input name="radiox" type="radio" value="radio1"/></pre>	
15	<pre> radio-set <input checked="" name="id1" type="radio" value="true"/> Yes</pre>	hidden -
16	<input name="id1" type="radio" value="false"/> No	checkhox - 🔽
17	<pre> drop down menu - <select name="select"></select></pre>	
18	<pre><option>cherry</option></pre>	radio - 🔾
19	<pre><option>orange</option></pre>	radio set Ver No
20	<pre><option>apple</option></pre>	ladoset © les O llo
21	-	drop down menu - cherry v
22	<pre> list menu - <select name="select2" size="3"></select></pre>	cherry A
23	<pre><option>cherry</option></pre>	clienty A
24	<pre><option>orange</option></pre>	orange
25	<pre><option>apple</option></pre>	list menu - apple V
26		image-Submit Queny
27	<pre> image-<input name="image" src="xulfr.gif" type="image"/></pre>	mage-subine query
28	<pre> submit - <input type="submit" value="Submit"/></pre>	submit - Submit
29	-	
30		
31	-	
32		
33	L	



Anatomy of a Hyper Text Markup Language Document:

• <u>A complete form example</u>:

```
<form name="myform" id="myform" action="page.html" onSubmit="return
fun()">
<br>button - <input type="button" value="Send" onclick="myfunction()">
<br>text - <input type="text" name="mytext" value="default text">
 <br> textarea - <textarea name="textarea">content</textarea>
<br>br>password - <input type="password" name="pass" value="12345">
<br>hidden - <input type="hidden" value="code">
<br>checkbox - <input type="checkbox" name="cbl" value="true" checked>
 <br>radio - <input type="radio" name="radiox" value="radio1">
<br>radio-set <input type="radio" name="id1" value="true" checked> Yes
<input type="radio" name="id1" value="false"> No
<br>drop down menu - <select name="select">
 <option>orange</option>
 <option>apple</option>
<br>list menu - <select name="select2" size="3">
 <option>orange</option>
 <option>apple</option>
</select>
<br>image-<input type="image" name="image" src="xulfr.gif">
<br>submit - <input type="submit" value="Submit">
```

107



Anatomy of a Hyper Text Markup Language Document: <u>HTML Tables</u>:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy


109

□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

TAG	Description
table	Defines a new table
tr	Defines a new row
td	Defines a new column
•••	





110

□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

TAG	Description
width & heigh	Defines table's dimensions
border	Defines border's size
cellspacing cellpading	Defines a distance betweeen cell and elements
•••	





111

□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

TAG	Description
bgcolor	Defines background color
bordercolor	Defines table's border color
borderstyle	Defines table's border style
•••	





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

o body section - tables:





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>







□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

```
o <u>body section - tables</u>:
```

```
alphabetagama
```

deltaepsilonzeta



□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

alpha	beta	gama
delta	epsilon	zeta
eta	theta	giota





Anatomy of a Hyper Text Markup Language Document:





□ Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

```
o <u>body section – tables (alignments)</u>:
```

 programming interpreted soripting < <

```
C
 c++
 java
php
 python
 matlab
javascript
 ...
 ...
```



Anatomy of a Hyper Text Markup Language Document:

<body>

This is my first page !!!

</body>

```
o <u>body section – tables (alignments)</u>:
```

programming interpreted soripting

```
C
 c++
 java
php
 python
 matlab
jayascript
 ...
 ...
```



Example of a complete HTML Form using Tables

<form name='registration' onSubmit="return formValidation();"> User id: <input type="text" name="userid" size="30" /> Password: <input type="password" name="passid" size="30" /> Name: <input type="text" name="username" size="30" /> Address: <input type="text" name="address" size="30" /> Country: <select name="country"> <option selected="" value="Default">(Please select a country)</option> <option value="GR">Greece</option> <option value="DE">Germany</option> <option value="IT">Italy</option> <option value="FR">France</option> <option value="SP">Spain</option> <option value="EN">England</option> <option value="IR">Ireland</option> <option value="SW">Sweden</option> <option value="FI">Finland</option> <option value="NO">Norway</option> </select> ZIP Code: <input type="text" name="zip" size="30" /> Email: <input type="text" name="email" size="30" /> Gender: <input type="radio" name="mgender" value="Male" />Male <input type="radio" name="fgender" value="Female" />Female Language: <input type="checkbox" name="en" value="en" checked /> English <input type="checkbox" name="ot" value="ot" /> Other About: <textarea name="desg" id="desg" rows="1" cols="10" id="myTextArea"></textarea> <input type="reset" name="reset" value="Reset" /><input type="submit" name="submit" value="Submit" /> </form>



Example of a complete HTML Form using Tables

Registration Form

User id:	alphanumeric
Password:	•••••
Name:	myName
Address:	myAddress
Country:	Greece v
ZIP Code:	123456
Email:	me@somewhere.com
Gender:	● Male ○ Female
Language:	\blacksquare English \square Other
About:	Further Information
Reset	Submit







- ✓ Client-side Scripting language
- ✓ Added to HTML documents
- Web-Browser Access and Controls HTML objects through scripts



- ✓ Client-side Scripting language
- ✓ Added to HTML documents
- Web-Browser Access and Controls HTML objects through scripts



- ✓ Script controls browser's actions
- ✓ Client Interaction
- ✓ Cookies
- ✓ Content Manipulation





JavaScript

Manipulating Browser Objects
 <u>Window</u>

... functionalities regarding browser's window:

<u>Appearance</u>

✓ moveTo() /ScrollTo()/resizeTo()✓ moveBy()/ScrollBy()/ResizeBy()



JavaScript

Manipulating Browser Objects
 <u>Window</u>

... functionalities regarding browser's window:
<u>Event Actions</u>
(on+ Load/Focus/Move/...etc)
✓ alert/confirm/prompt/...
✓ open/close



JavaScript

✓ Manipulating Browser Objects

o <u>Window</u>

1. document

... functionalities about

page contents

JavaScript

✓ Manipulating Browser Objects

- o <u>Window</u>
 - 1. document

2. frame

an array-like object, which represents all frame elements in the current window



JavaScript

- ✓ Manipulating Browser Objects
 - o <u>Window</u>
 - 1. document
 - 2. frame
 - 3. location

information about host and
its name, the path etc...
- location.href(...)

JavaScript

✓ Manipulating Browser Objects

- o <u>Window</u>
 - 1. document
 - 2. frame
 - 3. location
 - 4. history

information about the visited URLs concerning history (previous/current/next/back()/ forward()...etc)

JavaScript





- ✓ Manipulating Browser Objects
 - o <u>Window</u>
 - 1. document
 - a. <u>form</u>
 - button/submit/reset
 - text/textarea/password
 - checkbox/radio/select/hidden



□ JavaScript – Hands on Craft (Form Validation Example)



</html>



134

□ JavaScript – Hands on Craft (Form Validation Example)

<html lang="en"></html>
<head></head>
<meta charset="utf-8"/>
<title>JavaScript Form Validation using a sample registration form</title>
k rel='stylesheet' href='jg-form-validation.ggg' type='text/ggg' />
<pre><script src="sample-registration-form-validation.js"></script></pre>
<pre><body onload="document.registration.userid.focus();" onreset="return formEraseNotification();"></body></pre>
<h1 align="center">Registration Form</h1>
<pre></pre>
dlign="right">User id:
input type="text" name="userid" size="30" />
<pre>ctr>Dasewowd</pre>
<pre></pre>
ctr>Name:
iditions right "Address
<pre>/// ing fight function // data // ing // // // // // // // // // // // // //</pre>
<pre>classing or open one of the size of t</pre>
<pre></pre>
<pre><pre>conting selected="" value="Default">(Diase select a country)</pre></pre>
<pre>contion selected value selection > </pre>
<pre>continue of steele of steele statement of the ste</pre>
<pre>coption value= II = voltanty / option> </pre>
<pre>coption value= IT = "Totaly < potion"></pre>
Continu value= IR "France (option>
<pre>coption value= sp same options coption value= coptions coption value= coption value= coptions coption value= coption value= coptions coption value= coptions coption value= coptions coption value= coptions coption value= coption value= coptions coption value= coption value= coptions coption value= coptions coption value= coption value=</pre>
<pre>coption value= im /ingland/option/ coption value= The Neoling and /option/ coption</pre>
<pre>contion value= if SPIPSHode/option></pre>
<pre>coption value= sw sweeter/option> coption value= Transformer(spin)</pre>
<pre>contion value= r1 = rinandy/option> contion value= r10" Memory (option> </pre>
Addright Fight Sife Oue () do
Advinute light function and a second and a second and a second a s
Carry a light cype text hame email size 30 ////dx//i/
Addright Fight Mender, and "mender" value "Male"
<pre>classing type ratio name fifteener is a second s second second sec</pre>
Chipte type rate mane and the remain formate formate for the first state in the state of the sta
Concentration right Francisco (Concentration in the sheet of Concentration in the sheet of Conce
Construct type (mexanized and en value en checked / inglish
Single type (methods have of value of value of volter (/tdx/tl)
<pre>ctrxtd align= right xAbdut.v/tdx</pre>
<pre><td< th=""></td<></pre>
Substitute type= reset name= reset value= keset: /x/tdxtdxinput type= submit" name="submit" value="Submit" /x/td



□ JavaScript – Hands on Craft (Form Validation Example)

h1 {

margin-left: 70px; -} form li { list-style: none; margin-bottom: 5px; 1 form ul li label{ float: left; clear: left; width: 70px; text-align: right; margin-right: 10px; font-family:Verdana, Arial, Helvetica, sans-serif; font-size:14px; } form ul li input, select, span { float: left; margin-bottom: 10px; } form textarea { float: left; width: 200px; height: 15px; 1

[type="submit"] { clear: left; margin: 20px 0 0 230px; font-size:18px } [type="reset"] { clear: left; margin: 20px 0 0 230px; font-size:18px ł p { margin-left: 70px; font-weight: bold; ł table .absorbing-column { width: 100%; } #myTextArea { min-width: 10px; min-height: 5px; ł

sampleregistration-formvalidation.js

□ JavaScript – Hands on Craft (Form Validation Example)

```
function formValidation() {
    var uid = document.registration.userid;
    var passid = document.registration.passid;
    var uname = document.registration.username;
    var uadd = document.registration.address;
    var ucountry = document.registration.country;
    var uzip = document.registration.zip;
    var uemail = document.registration.email;
    var umgender = document.registration.mgender;
    var ufgender = document.registration.fmgender;
    if(userid_validation(uid,5,12)){
        if(passid validation(passid,7,12)) {
            if(allLetter(uname)) {
                if (alphanumeric (uadd)) {
                    if(countryselect(ucountry)) {
                        if(allnumeric(uzip)) {
                             if(ValidateEmail(uemail)) {
                                 if(validgender(umgender, ufgender)) {
                                 }
    return false;
```

}

sampleregistration-formvalidation.js

137

□ JavaScript – Hands on Craft (Form Validation Example)

```
function userid validation(uid,mx,my) {
    var uid len = uid.value.length;
    if (uid_len == 0 || uid_len \succ my || uid_len < mx) {
        alert ("User Id should not be empty / length be between "+mx+" to "+my);
        uid.focus();
        return false;
    3
    return true;
}
function passid validation(passid,mx,my) {
    var passid len = passid.value.length;
    if (passid len == 0 ||passid len >= my || passid len < mx) {
        alert ("Password should not be empty / length be between "+mx+" to "+my);
        passid.focus();
        return false;
    ι
    return true;
3
function allLetter(uname) {
    var letters = /^[A-Za-z]+$/;
    if(uname.value.match(letters)) {
        return true;
    3
    else{
        alert ('Username must have alphabet characters only');
        uname.focus();
        return false;
3
```

sampleregistration-formvalidation.js

□ JavaScript – Hands on Craft (Form Validation Example)

```
function alphanumeric (uadd) {
    var letters = /^{[0-9a-zA-Z]+\$/};
    if(uadd.value.match(letters)) {
        return true;
    l
    else{
        alert ('User address must have alphanumeric characters only');
        uadd.focus();
        return false:
3
function countryselect (ucountry) {
    if(ucountry.value == "Default") {
        alert ('Select your country from the list');
        ucountry.focus();
        return false;
    3
    else{
        return true;
    1
3
function allnumeric(uzip) {
    var numbers = /^{[0-9]+$/};
    if(uzip.value.match(numbers)) {
        return true;
    3
```

else{

```
alert('ZIP code must have numeric characters only');
uzip.focus();
return false;
```

sampleregistration-formvalidation.js

□ JavaScript – Hands on Craft (Form Validation Example)

```
function ValidateEmail(uemail){
    var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
    if(uemail.value.match(mailformat)) {
        return true;
    }
    else{
        alert("You have entered an invalid email address!");
        uemail.focus();
        return false;
    }
}
function validgender(umgender,ufgender){
        x=0;
    if(umgender.checked){
            x++;
    }
```

x++;

```
if(x==0){
```

if(ufgender.checked) {

```
alert('Select Male/Female');
umgender.focus();
return false;
```

} else{

```
alert('Form Succesfully Submitted');
window.location.reload()
return true;
```

```
}
```

```
function formEraseNotification() {
```

```
var x = confirm("Are you sure you want to erase form?");
if (x)
    return true;
else
    return false;
return true;
```





PHP BASICS



PHP: PHP **H**ypertext **P**reprocessor

✓ Used for Dynamic Modification of the content of a web page (dynamic web pages).





- ✓ Used for Dynamic Modification of the content of a web page (dynamic web pages).
- ✓ Manipulation of data from HTML forms





- ✓ Used for Dynamic Modification of the content of a web page (dynamic web pages).
- ✓ Manipulation of data from HTML forms
- ✓ Accessing DataBases





- ✓ Used for Dynamic Modification of the content of a web page (dynamic web pages).
- ✓ Manipulation of data from HTML forms
- ✓ Accessing DataBases
- ✓ Server-Side Scripting (non-visible source code)




PHP: PHP Hypertext **P**reprocessor

- ✓ Used for Dynamic Modification of the content of a web page (dynamic web pages).
- ✓ Manipulation of data from HTML forms
- ✓ Accessing DataBases
- ✓ Server-Side Scripting (non-visible source code)
- ✓ Results can be rendered to browser through HTML



PHP: PHP **H**ypertext **P**reprocessor





PHP: PHP **H**ypertext **P**reprocessor







PHP: Programming Basics

- ✓ <?php ...place your code here... ?>
- ✓ Type declaration before using a variable is not prerequisite:

```
$my_num=3.14;
$my alphanumeric="str123str"; etc...
```

✓ Variables can be displayed through "echo":

```
echo "number: " .$my_num;
echo "string: " .$my_alphanumeric;
```



150

- ✓ Every statement ends with semicolon:
 - \$a = 5;
 - \$a = function();
 - \$a = (\$b = 5);
 - \$a++; ++\$a;
 - \$a += 3;



- ✓ <?php ...place your code here... ?>
 ✓ Operators:
 - Arithmethic: +,-, * , %
 - Setting variable: =
 - Bit: &, |, ^ , ~ , <>
 - Comparison: == , === , != , != , <= , >=





```
<?php ...place your code here... ?>

Iteration Sequence Statements:
```

```
<?php
//----- while loop ------
$a=0;
while($a<10){print $a; $a++;}
```



PHP: Programming Basics

✓ Iteration Sequence Statements:

```
<?php
    //----- while loop ------
    $a=0;
    while($a<10){print $a; $a++;}
    //----- do while loop ------
    $i = 0;
    do {print $i;} while ($i > 0);
    }
}
```



PHP: Programming Basics

< <?php ...place your code here... ?>

✓ Iteration Sequence Statements:

<?php
 //----- while loop ----- \$a=0;
 while(\$a<10){print \$a; \$a++;}
 //----- do while loop ----- \$i = 0;
 do {print \$i;} while (\$i > 0);
 //----- for loop ----- for(\$i = 1; \$i <= 10; \$i++){print \$i;}
}</pre>



PHP: Programming Basics

< <?php ...place your code here... ?>

✓ Iteration Sequence Statements:

```
<?php
    //----- while loop ------
    $a=0;
    while($a<10){print $a; $a++;}
    //----- do while loop ------
    $i = 0;
    do {print $i;} while ($i > 0);
    //----- for loop ------
    for($i = 1; $i <= 10; $i++){print $i;}
    //----- foreach loop ------
    $arr = array(1, 2, 3, 4);
    foreach($arr as $value){echo $value;}
    ?>
```



PHP: Programming Basics

< <?php ...place your code here... ?>

✓ Iteration Sequence Statements:

```
<?php
    //----- while loop ------
    $a=0;
    while($a<10){print $a; $a++;}
    //----- do while loop ------
    $i = 0;
    do {print $i;} while ($i > 0);
    //----- for loop ------
    for($i = 1; $i <= 10; $i++){print $i;}
    //----- foreach loop ------
    $arr = array(1, 2, 3, 4);
    foreach($arr as $value){echo $value;}
    ?>
```



PHP: Programming Basics

```
< <?php ...place your code here... ?>
</ Functions:
</?php
$name="joe";
$target="prof";
$str= "what Joe wanna be? <br>".strcnt($name,$target);
function strcnt($x,$y){
    print "-->".$x."wanna be a".$y."<br>";
}
```







- ✓ <?php ...place your code here... ?>

 Data Sanitization
- Example: stripslashes ()
- \$str = "Is your name O\'reilly?";
- // Outputs: Is your name O'reilly?
- echo stripslashes(\$str);



```
</?php ...place your code here... ?>
Data Sanitization
Example: addslashes()
```

```
$str = "O'Reilly?";
```

```
eval("echo '" . addslashes($str) . "';");
```



PHP: Programming Basics

```
✓ <?php ...place your code here... ?>
<u>Data Sanitization</u>
```

```
Example: get_magic_quotes_gpc()
```

```
if (get_magic_quotes_gpc()) {
    $data = stripslashes($_POST['data']);
```

```
}
```

```
else {
```

```
$data = $ POST['data'];
```

```
}
```

http://php.net/manual/en/function.get-magic-quotes-gpc.php

</html>

PHP: Authenticating User Account

```
<html>
<head>
<title>LogIn As Administrator</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>
<body>
<form align=left name="theform" method="post" action="em intro page.htm">
<input type="submit" value="BACK"></form>
<h2 align=center><b><i>Administration</i></b></h2>
<body background=img24.jpg>
<body text=white>
<div align="center">
   <form name="form" method="POST" action="em admin auth.php">
   d>USERNAME:</b>
      <input type="text" name="adminUsername"/>
     \langle t,r \rangle
      PASSWORD:</b>
      input type="password" name="adminPassword"/>
     \langle tr \rangle
       
      <label>
        <input type="submit" name="Submit" value="LogIn" />
      </label>
     </form>
</div>
</body>
```



<html>

?>



165

PHP: Authenticating User Account

```
<head><title>USER AUTHENTICATION</title></head>
<body text=black></body>
</html>
<?php
if (empty($ POST['adminUsername']) || empty($ POST['adminPassword']))
   echo "test1";
   echo "<script language=\"JavaScript\" type=\"text/javascript\"> alert('Submit both username and password'); </script>";
   echo "<meta http-equiv='refresh' content='0; url=em_admin_auth.htm'>";
    exit();
else
   echo "test2";
   $username_from_form = $_POST['adminUsername'];
   $password_from_form = $_POST['adminPassword'];
    $hashed username usr = sha1($username from form);
    $hashed password usr = sha1($password from form);
    $dbusnm='12dea96fec20593566ab75692c9949596833adc9';//user
    $dbpswd='9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684';//polenakis
    if (strcmp($hashed username usr, $dbusnm) != 0 || strcmp($hashed password usr, $dbpswd) != 0)
        echo "<script language=\"JavaScript\" type=\"text/javascript\"> alert('the combination of username and password is not correct'); </script>";
        echo "<meta http-equiv='refresh' content='0; url=em admin auth.htm'>";
        exit();
    else
        echo "Mr/s ".$username from form."Welcome";
        session start();
        $ SESSION['is administrator logged in'] = true;
        echo "<meta http-equiv='refresh' content='0; url=index.html'>";
```



□ PHP– Hands on Craft (Interaction with DB in MySql)





167

PHP: Connection to Database

PHP BASICS

<form name='registration' action="insert into db A.php" method="POST" onSubmit="return formValidation();"> User id: <input type="text" name="userid" size="30" /> Password: <input type="password" name="passid" size="30" /> Name: <input type="text" name="username" size="30" /> Address: input type="text" name="address" size="30" /> Country: select name="country"> <option selected="" value="Default">(Please select a country)</prion> <option value="GR">Greece</option> <option value="DE">Germany</option> <option value="IT">Italy</option> <option value="FR">France</option> <option value="SP">Spain</option> <option value="EN">England</option> Registration.html <option value="IR">Ireland</option> <option value="SW">Sweden</option> <option value="FI">Finland</option> <option value="NO">Norway</option> </select> ZIP Code: <input type="text" name="zip" size="30" /> Email: <input type="text" name="email" size="30" /> About: <textarea name="desc" id="desc" rows="1" cols="10" id="myTextArea"></textarea> input type="reset" name="reset" value="Reset" />/td>input type="submit" name="submit" value="Submit" />/td> $\langle tr \rangle$ </form>



PHP: Connection to Database

PHP BASICS

<form action="insert_into_db A.php" method="POST" name="registration" onsubmit="return formValidation();"></form>			
User id:			
input type="text" name"userid")ize="30" />			
Password:			
input type="password" name"passid" ize="30" />			
Name:			
input type="text" name="username")qize="30"/>			
Address:			
input type="text" name="address"aize="3%" />/td//tr>			
Country:			
select name"country"			
<pre><option selected="" value="Default">(Please select a country)</option></pre>			
<pre><option value="GR">Greece</option></pre>			
<pre><option value="DE">Germany</option></pre>			
<pre><option value="IT">Italy</option></pre>			
<pre><option value="FR">France</option></pre>			
<pre><option value="SP">Spain</option></pre>			
<pre><option value="EN">England</option></pre>			
<pre><option value="IR">Ireland</option></pre>			
<pre><option value="SW">Sweden</option></pre>			
<pre><option value="FI">Finland</option></pre>			
<pre><option value="NO">Norway</option></pre>			
ZIP Code:			
<input name="zip" size="30" type="text"/> >/td>			
Email:			
<input bize="30" name="email" type="text"/>			
About:			
textarea name="desg" id="desg" rows="1" cols="10" id="myTextArea">			
<input name="reset" type="reset" value="Reset"/> <input name="submit" type="submit" value="Submit"/>			









PHP: Connection to Database (MySQLi)

PHP BASICS

Instruction		
connect to DB	<pre>\$con = new mysqli(`localhost', `user',`admin',`project_db');</pre>	<pre>\$con = mysqli_connect(`localhost', `user', `admin', `project_db');</pre>
select DB	<pre>\$con->select_db(project_db);</pre>	<pre>mysqli_select_db(\$con, `project_db');</pre>
query execution	<pre>\$select_data= \$con->query();</pre>	<pre>\$select_data= mysqli_query(\$con,query);</pre>
fetch results	<pre>\$tuple= \$result->fetch_array() ;</pre>	<pre>\$tuple=mysqli_fetch_array(\$select_data);</pre>
close connection	<pre>\$con->close();</pre>	<pre>mysqli_close();</pre>