

COSC 4351 Fall 2023

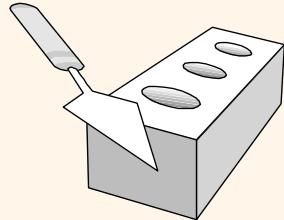
Software Engineering

M & W 4 to 5:30 PM

Prof. **Victoria Hilford**

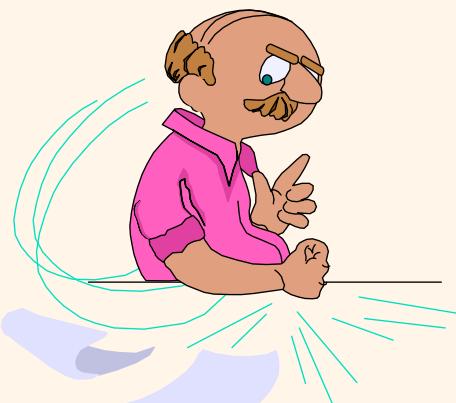
PLEASE TURN your webcam ON

NO CHATTING during LECTURE



COSC 4351

4 to 5:30



**PLEASE
LOG IN
CANVAS**

Youyi [A-L]

Kevin [M-Z]

Please close all other windows.

Search content Search [View content explorer](#) [Configure zyBook](#)

Showing activity for entire class [zyLabs](#) [Challenge](#) [Participation](#)

<input type="checkbox"/> 1. Introduction to Web Programming	█ 97%	█ 99%	▼	
<input type="checkbox"/> 2. HTML	█ 80%	█ 94%	█ 99%	▼
<input type="checkbox"/> 3. More HTML	█ 69%	█ 90%	█ 96%	▼
<input type="checkbox"/> 4. Basic CSS	█ 68%	█ 85%	█ 93%	▼
<input type="checkbox"/> 5. Advanced CSS	█ 62%	█ 83%	█ 92%	▼
<input type="checkbox"/> 6. Basic JavaScript	█ 6%	█ 23%	█ 28%	▼
<input type="checkbox"/> 7. JavaScript in the Browser	█ 2%	█ 5%	█ 9%	▼
<input type="checkbox"/> 8. Advanced JavaScript	█ 0%	█ 3%	█ 4%	▼
<input type="checkbox"/> 9. jQuery	█ 0%	█ 1%	█ 3%	▼

COSC 4351:
Fundamentals of
Software Engineering
Spring 2023 

[View activity and create a report](#)

1. Select chapters and sections in the table of contents.
2. Then select class and time options below.

Entire class [▼](#)

From: Select date Select time [▼](#) CST

Until: Feb 22nd, 2023 1:00 pm [▼](#) CST

[Stop viewing entire class](#)

All activity up until Feb 22nd, 2023 at 1:00 pm CST will be downloaded.

Include data on time spent in chapters, sections, and on activities

[Download report](#)

You must select at least one section from the table of contents to download a report.

Class analytics (beta feature)

View time spent, completion, and timeline of activity for your entire class or individual students on chapters, sections, and activities.

09.27.2023 (W 4 to 5:30)	Estimating and Planning: 1. Estimating Techniques 2015.pdf 2. How do you estimate on an Agile Project 2015.pdf	Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks WEBCANVAS APPS	WEB APP Papers Summary (1 Page) CANVAS Assignment
10.02.2023 (M 4 to 5:30)	(12)	Lecture 4: Estimating and Planning Tutorials 4 COCOMO and MS Project	Estimating and Planning Papers Summary (1 Page) CANVAS Assignment
10.04.2023 (W 4 to 5:30)	(13)	EXAM 2 REVIEW (CANVAS) (ZyBook)	Download ZyBook: Sections 5-9
10.09.2023 (M 4 to 5:30)	Optional (14)		Q & A Set 2 topics
10.11.2023 (W 4 to 5:30)	(15)		EXAM 2 (CANVAS) (ZyBook)

Class 11

Tutorial 3 on WEB APPS

09.27.2023
(W 4 to 5:30)

(11)

Estimating and Planning:

1. Estimating Techniques
2015.pdf

2. How do you estimate
on an Agile project
2015.pdf

Tutorials 3 on C# Visual
Studio IDE MVC,
PYTHON PyCharm IDE
Django, PHP PhpStorm
IDE Zend frameworks
WE CANVAS APPS

WEB APP Papers

Summary

(1 Page)

CANVAS
Assignment

From 4:00 to 4:10 – 10 minutes.

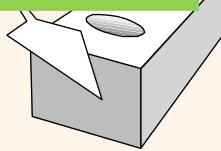
09.27.2023 (W 4 to 5:30) (11)	Estimating and Planning: 1. Estimating Techniques 2015.pdf 2. How do you estimate on an Agile project 2015.pdf	 Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks WECANVAS APPS	 WEB APP Papers Summary (1 Page) CANVAS Assignment		
---	--	--	---	--	--

CLASS PARTICIPATION 20 points 20% of Total + :

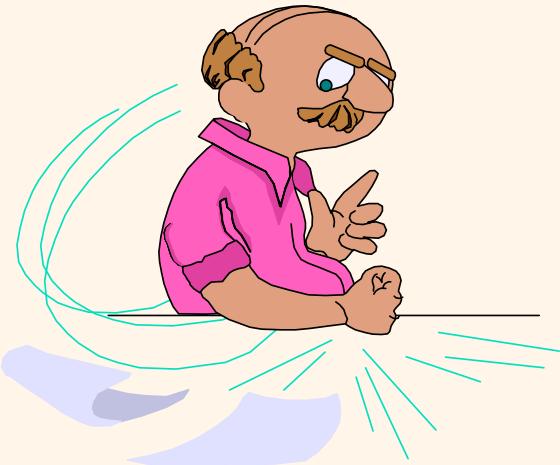
PASSWORD: IN TEAMS

BEGIN Class 11 Participation
CLASS PARTICIPATION 20% Module | Not available until Sep 27 at 4:00pm | Due Sep 27 at 4:10pm | 100 pts

From 4:10 to 4:30 – 20 minutes.



Internet/Web Applications with a database back-end



Database Application

Development

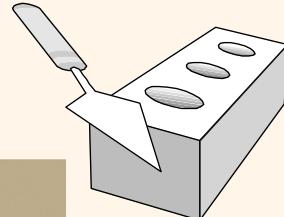
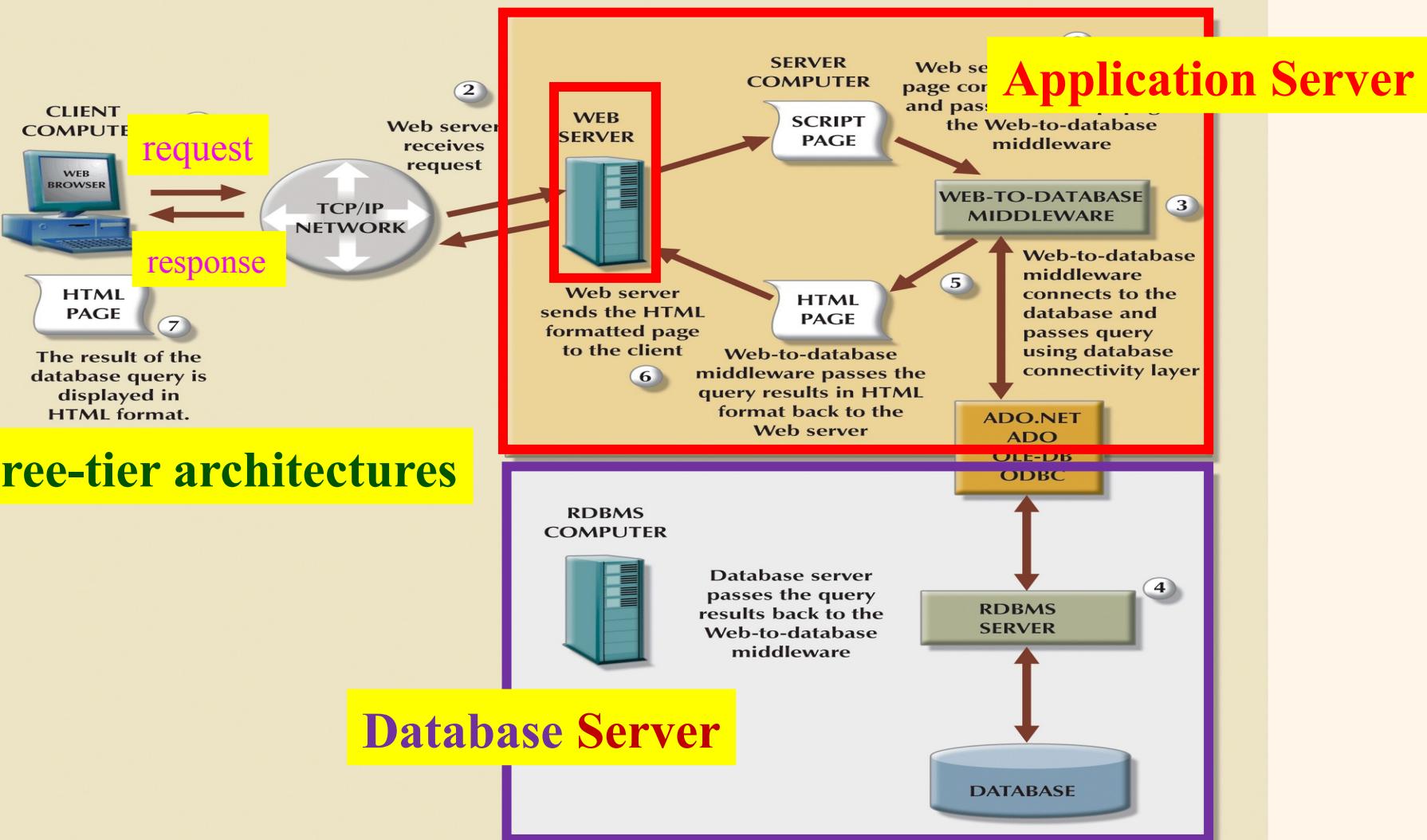


FIGURE
14.7

Web-to-database middleware



Three-tier architectures

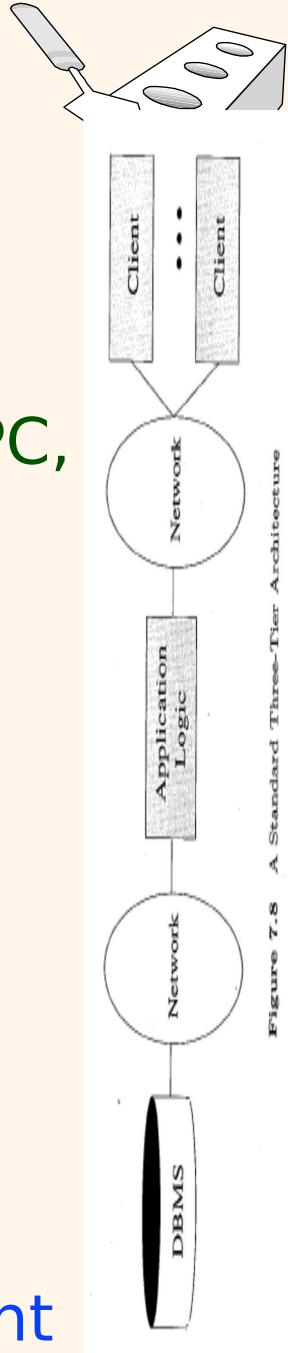
Database Server

MS SQL SERVER DBMS

Oracle DBMS

mySQL DBMS

The Three Layers (**MVC**)



Presentation tier

- Primary interface to the User
- Needs to adapt to different display devices (PC, PDA, cell phone, voice access?)

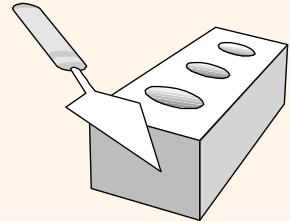
Middle tier

- Implements business logic (implements complex actions, maintains state between different steps of a workflow)
- Accesses different **data management systems**

Data management tier

- One or more standard **Database Management**

Example 1: Airline Reservations



Build a **system for making** Airline Reservations

What is done in the different tiers?

(M)

- **Database System**

- Airline info, available Seats, Customer info, etc.

(C)

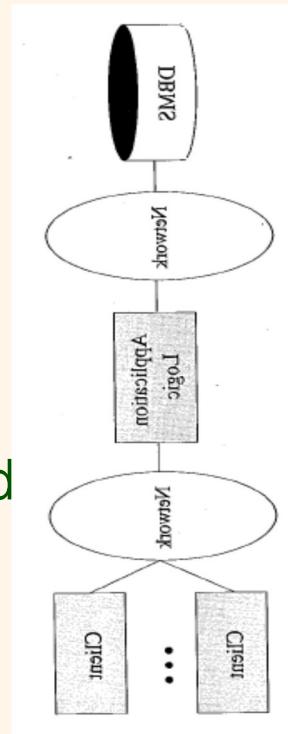
- **Application Server**

- Logic to make reservations, cancel reservations, add new airlines, etc.

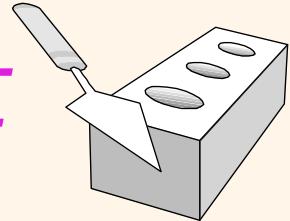
(V)

- **Client Program**

- Log in different **Users**, display forms (**Input Forms**)
Reports)



Example 2: Course Enrollment



Build a **system** using which Students can enroll in courses

❖ Database System^(M)

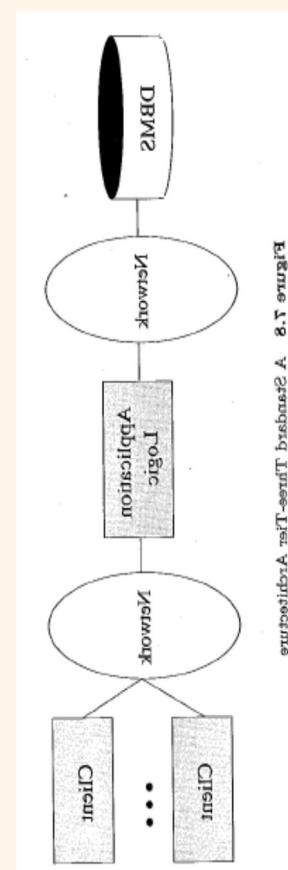
- Student info, Course info, Instructor info, Course availability, Pre-requisites, etc.

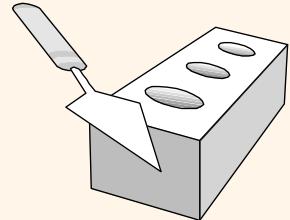
❖ Application Server^(C)

- Logic to add a course, drop a course, create a new course, etc.

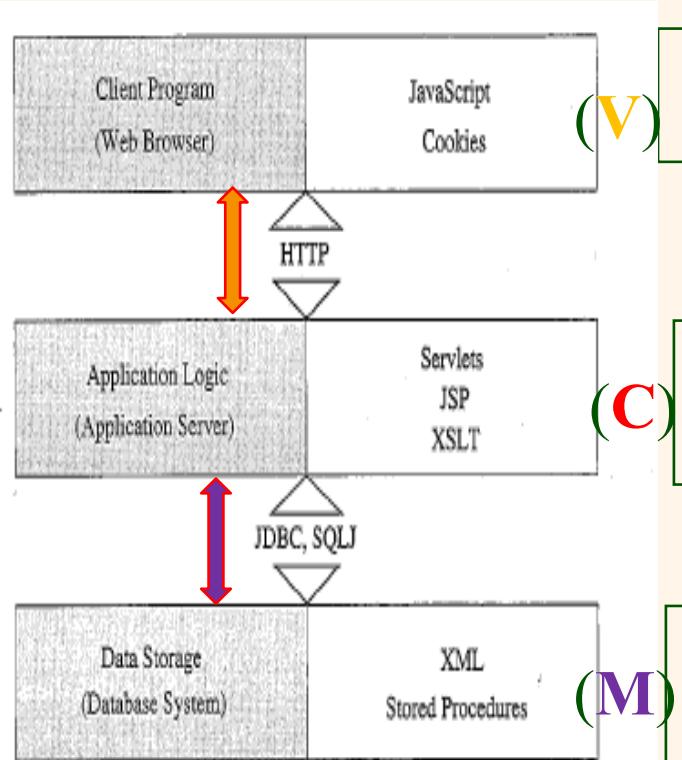
❖ Client Program^(V)

- Log in different **Users** (students, staff, faculty), display forms (**Input Forms**) and human readable output (**Output Reports**)





3-tier Technologies (MVC)



Client Program
(*Web Browser*)

Application Server
(*Tomcat, Apache, IIS*)

Database Server
(*DB2*)

HTML
Javascript
CSS *JQuery*
XSLT
Cookies

JSP
Servlets
CGI
ASP.NET
PHP

XML
Stored Procedures

Figure 7.9 Technologies for the Three Tiers

Web Database Application

Development

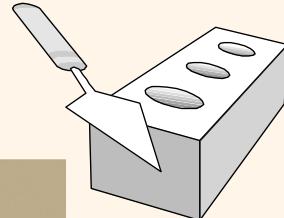
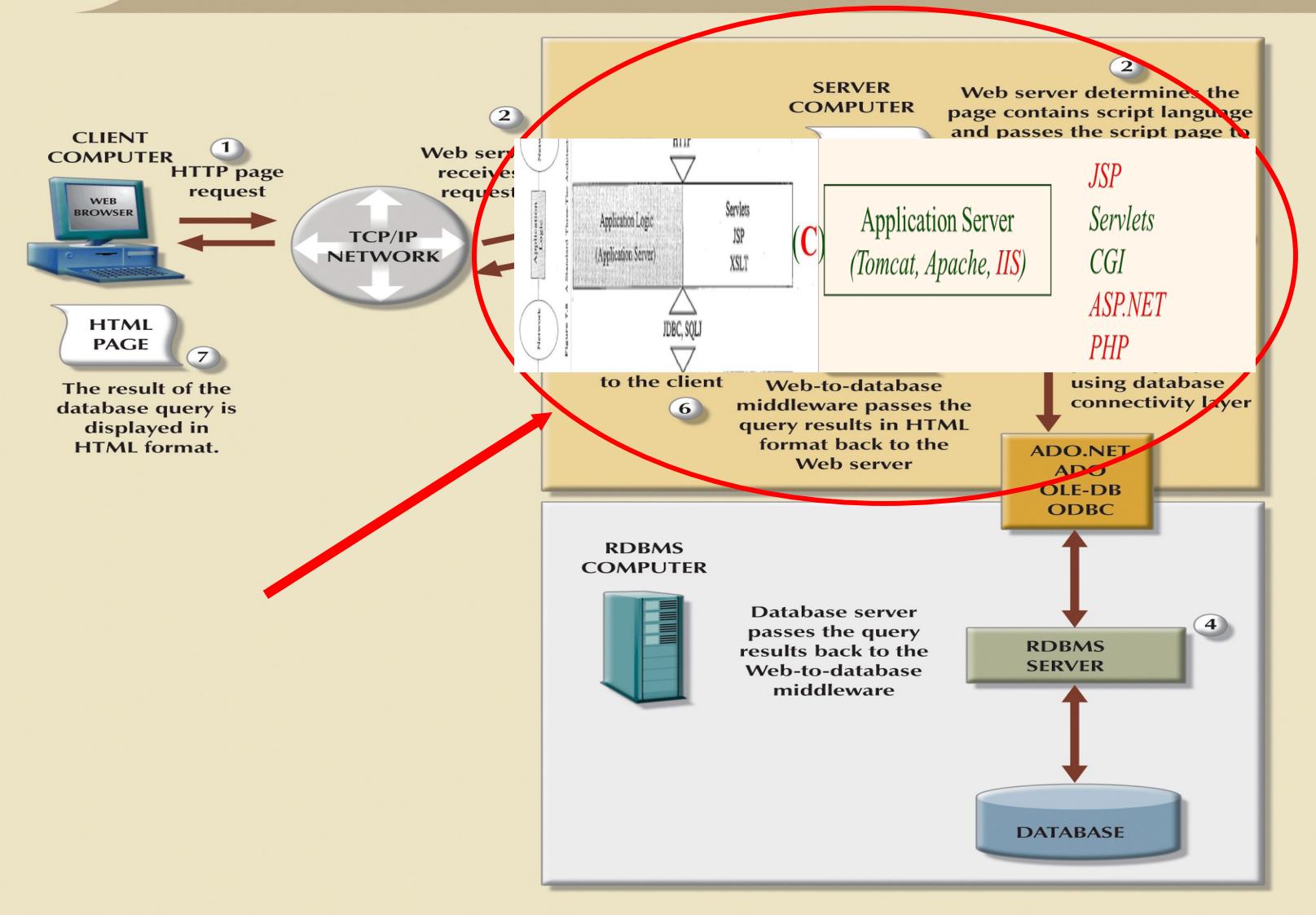
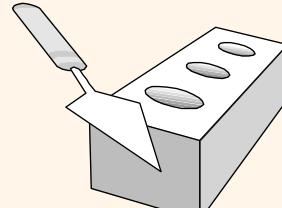


FIGURE
14.7

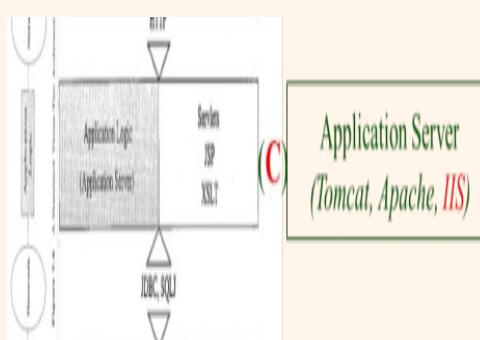
Web-to-database middleware



Functionality of the Middle Tier



- Encodes business logic **(C)**



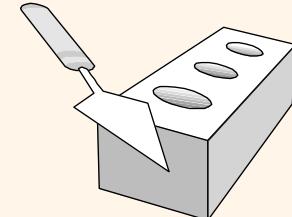
JSP
Servlets
CGI
ASP.NET
PHP

- Connects to **database system(s)** **(M)**

- Accepts **Form input** from the **Presentation tier**
- Generates **output** for the **Presentation tier**



CGI: Common Gateway Interface



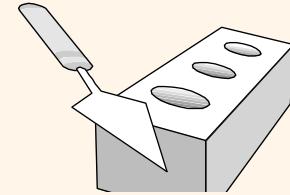
Goal: Transmit arguments from **HTML Forms** to Application programs running at the **middle tier**

Details of the actual **CGI** protocol unimportant
☞ libraries implement high-level interfaces

Disadvantages:

- The Application program is invoked in a new process at every invocation (remedy: **FastCGI**)
- No resource sharing between applications (e.g., **Database** connections)

ta How is the **HTML Form** specifying **the program** that will receive the **Form input parameters**



value = Send

value = "Clear



CGI: Example

- HTML form (**Input Form**) :

```
<form action="findbooks.cgi" method=POST>
    Type an author name:
    <input type="text" name="authorName">
    <input type="submit" value="Send">
    <input type="reset" value="Clear" form="">
</form>
```

- Perl code (**Output Report**) :

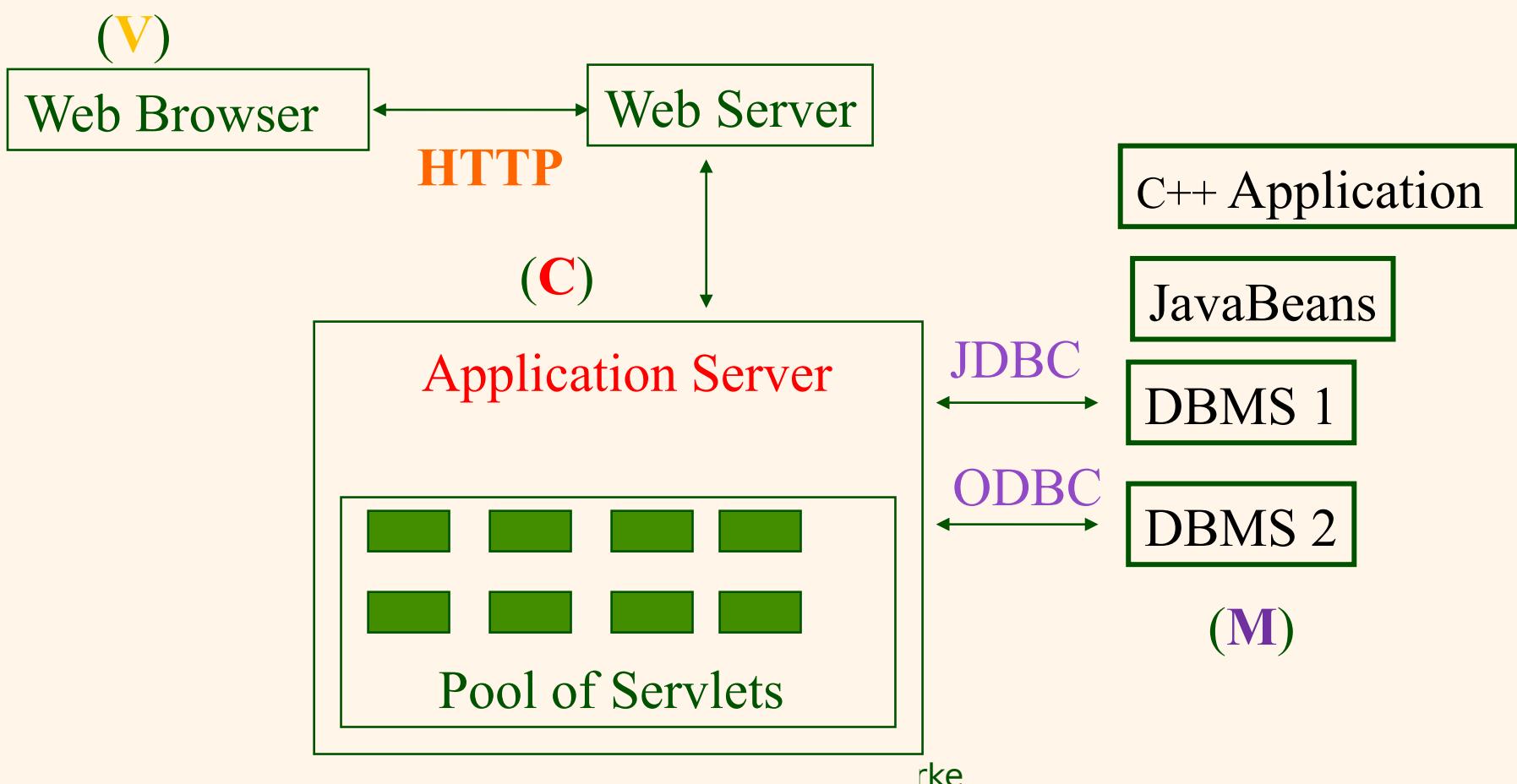
```
use CGI;
$dataIn=new CGI;
$dataIn->header();
$authorName=$dataIn->param('authorName');
print("<HTML><TITLE>Argument passing test</TITLE>");
print("The author name is " + $authorName);
print("</HTML>");
exit;
```

Output Report is in **HTML**

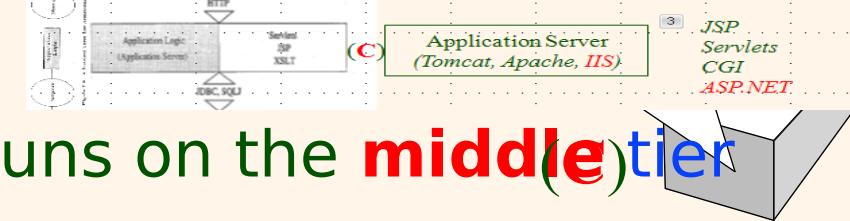
Application Server Process

Structure

- Main pool of threads of processes
- Manage connections(M)
- Enable access to heterogeneous data sou
- Other functionality such as **APIs** for **session management**



Servlets



- ❖ Java **Servlets**: Java code that runs on the **middle(e)tier**
 - Platform independent
 - Complete Java **API** available, including **JDBC**

Example:

```
import java.io.*;  
import java.servlet.*;  
import java.servlet.http.*;
```

JAVA first!

Produces HTML!

```
public class ServetTemplate extends HttpServlet  
{
```

```
    public void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException
```

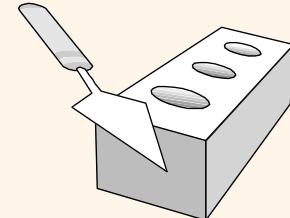
```
    {
```

```
        PrintWriter out=response.getWriter();  
        out.println("Hello World");
```

```
}
```

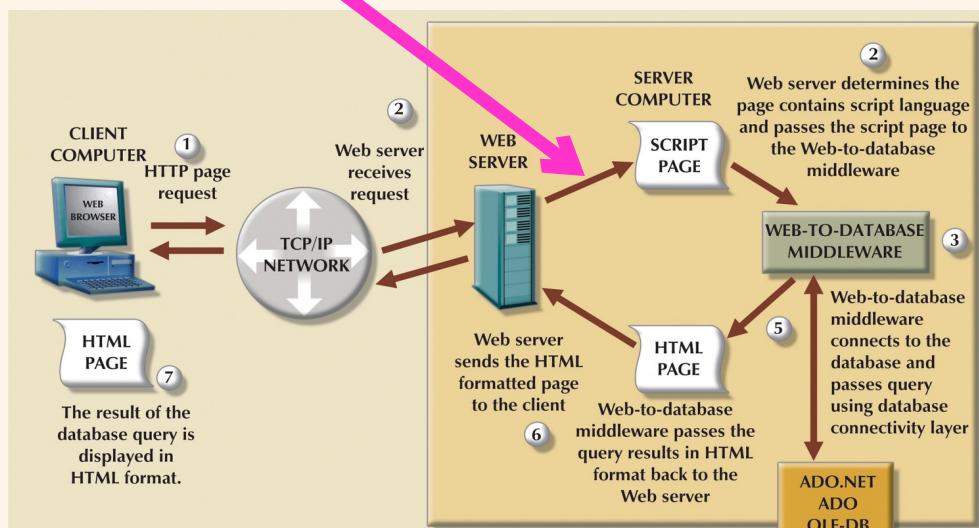
```
ta
```

Servlets

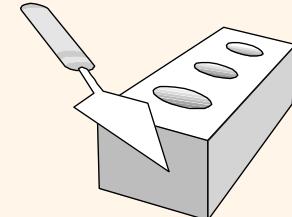


Life of a **Servlet**?

- **Web Server** forwards **request** to **Servlet** container
- Container creates servlet instance (calls **init()** method; deallocation time: calls **destroy()** method)
- Container calls **service()** method
 - **service()** calls **doGet()** for **HTTP GET** or **doPost()** for **HTTP POST**



Servlets: A Complete Example



```
public class ReadUserName extends HttpServlet  
{
```

```
    public void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException
```

```
    {  
        response.setContentType("text/html");  
        PrintWriter out=response.getWriter();  
        out.println("<HTML><BODY>\n <UL> \n" +  
                  "<LI>" + request.getParameter("userid") +  
                  "<LI>" + request.getParameter("password") + "\n" +  
                  "<UL>\n<BODY></HTML>");  
    }
```

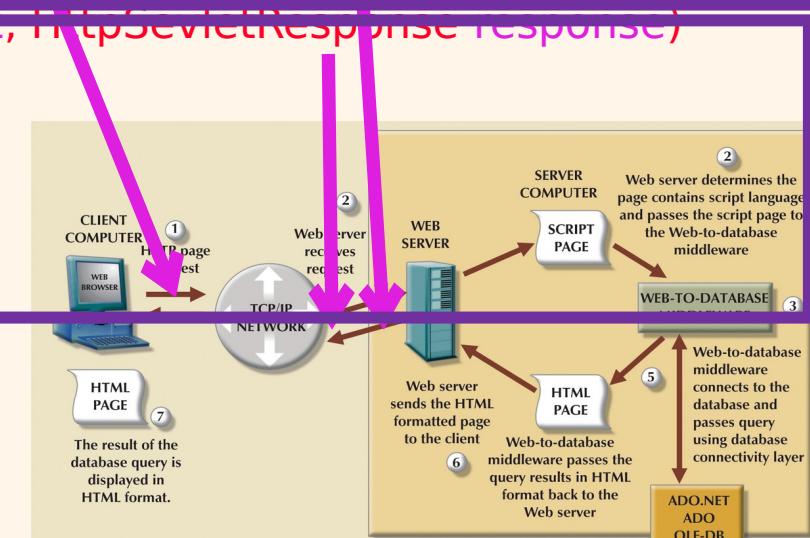
JAVA first!

Produces HTML!

Output Report is in HTML

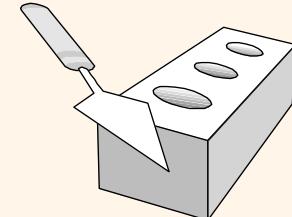
```
    public void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException
```

```
    {  
        doGet(request,response);  
    }
```



Servlets vs. Java Server

Pages (JSP)



❖ Servlets

- Generate **HTML** by writing it to the “**PrintWriter**” object **out**
- Code first, Webpage second (**HTML**)

JAVA first!

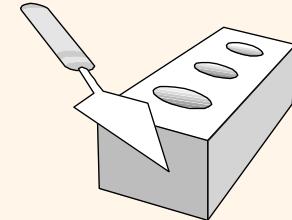
❖ Java Server Pages (JSP)

- Written in **HTML**, **Servlet**-like code embeds **HTML** first!
- Webpage first (**HTML**), Code second
- They are usually compiled into a **Servlet**

```
PrintWriter out=response.getWriter();
out.println("<HTML><BODY>\n <UL> \n" +
            "<LI>" + request.getParameter("userid") + "\n" +
            "<LI>" + request.getParameter("password") + "\n" +
            "<UL>\n<BODY></HTML>");
```

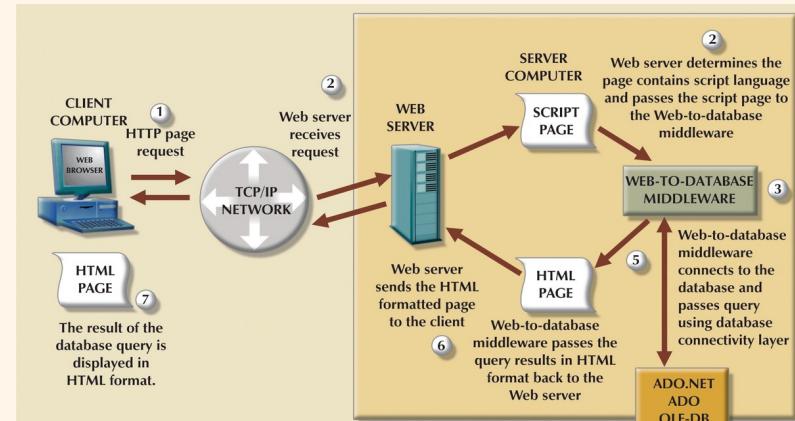
like

Java Server Pages (JSP)



❖ Java Server Pages

- Written in **HTML**, **JAVA** Code embedded in the **HTML**
- Webpage first, Code second **HTML** first!



Java Server Pages: Example



```
<html>
  <head><title>Welcome to B&N</title></head>
  <body>
    <h1>Welcome back!</h1>
    <% String name="NewUser";
      if (request.getParameter("username") != null)
      {
        name=request.getParameter("username");
      }
    %>
    You are logged on as user <%=name%>
  </body>
</html>
```

JAVA

JAVA

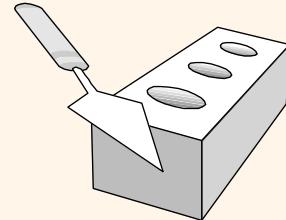
JSP to oracle 12c DBMS: A Complete Example

The screenshot shows the NetBeans IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard Java development toolbar with icons for file operations, search, and run.
- Projects Tab:** Shows a project named "JSP NetBeans 6.9.1 Web Application connecting to Oracle 12c JDBC". It contains Web Pages (META-INF, WEB-INF, index.jsp), Source Packages, Test Packages, and Libraries (ojdbc6.jar).
- Files Tab:** Currently viewing "index.jsp".
- Code Editor:** Displays JSP code connecting to an Oracle database and displaying results in an HTML table.

```
<html>
<body>
    <TABLE>
        <TR>
            <TH>Aircraft Type</TH>
            <TH>Description</TH>
            <TH>Serial No</TH>
        </TR>
        <%--  Setup the Output Report HTML View --%>
        <TABLE BORDER="1" BGCOLOR="CCFFFF" width='50%' cellspacing='1'>
            <TR>
                <TH bgcolor='#DAA520'> <font size='2'></font></TH>
                <TH bgcolor='#DAA520'> <font size='2'></font></TH>
                <TH bgcolor='#DAA520'><font size='2'></font></TH>
            </TR>
            <% while (resultset.next()) { %>
            <TR>
                <TD> <font size='2'></font><center><%= resultset.getString(1)></center></TD>
                <TD> <font size='2'></font><center><%= resultset.getString(2)></center></TD>
                <TD> <font size='2'></font><center><%= resultset.getString(3)></center></TD>
            </TR>
            <% } %>
        </TABLE>
    </body>
</html>
```
- Search Bar:** Search (Ctrl+I) at the top right.
- Java Annotations:** Two yellow boxes with the word "JAVA" are overlaid on the code editor area, highlighting specific sections of the code.
- Status Bar:** Shows "50 | 1" and "INS" at the bottom right.

JSP to oracle 12c DBMS: A Complete Example



select * from VHILFORD.AI... x

#	AIRCRAFTTYPE	ADESCRIPTION	SEATCAPACITY
1	ATP	Advanced Turbo Prop	48
2	DC9	McDonnel Douglas Jet	120
3	737	Boeing 737-300 Jet	300

DATABASE TABLE

AIRCRAFT

http://localhost:8080/

File Edit View Favorites Tools Help

Favorites Suggested Sites Free Hotmail Web Slice Gallery Bing Traffic Customize Links

JSP Page

JSP Page connecting to Oracle 12c Table!

Aircraft Type	Description	Seating Capacity
ATP	Advanced Turbo Prop	48
DC9	McDonnel Douglas Jet	120
737	Boeing 737-300 Jet	300

Done Local intranet 125%

WEB PAGE

JSP Page connecting to Oracle 12c Table!

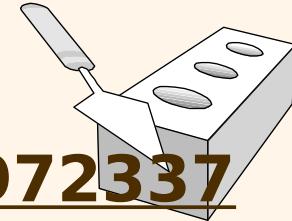
Aircraft Type	Description	Seating Capacity
ATP	Advanced Turbo Prop	38
DC9	McDonnel Douglas Jet	110
737	Boeing 737-300 Jet	290
727	Boeing 727-200	190

select * from VHILFORD.AIRCRAFT

#	AIRCRAFTTYPE	ADESCRIPTION	SEATCAPACITY
1	ATP	Advanced Turbo Prop	48
2	DC9	McDonnel Douglas Jet	120
3	737	Boeing 737-300 Jet	300

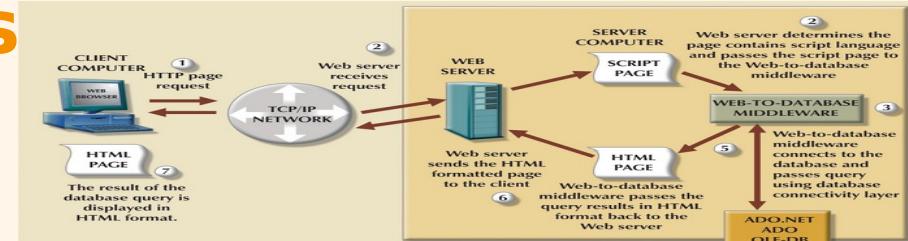
NetBeans IDE has Apache Tomcat built in to serve this web page

Active Server Pages (ASP)

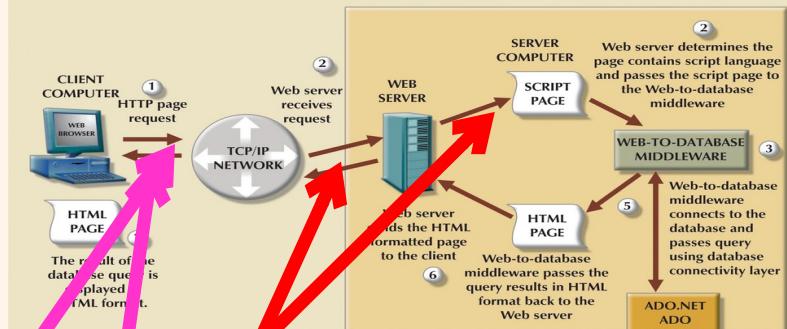


<http://msdn.microsoft.com/en-us/library/ms972337.aspx>

- ❖ Active Server Pages
- ❖ A series of **objects** and components that are executed *on the web server IIS*
- ❖ Uses a suite of technologies that allows **dynamically-generated** content
- ❖ Control of **how content** is generated *from the Server to the Browsers*

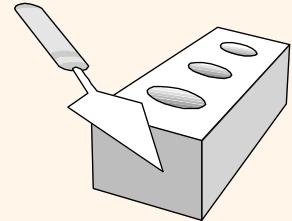


ASP Page Execution



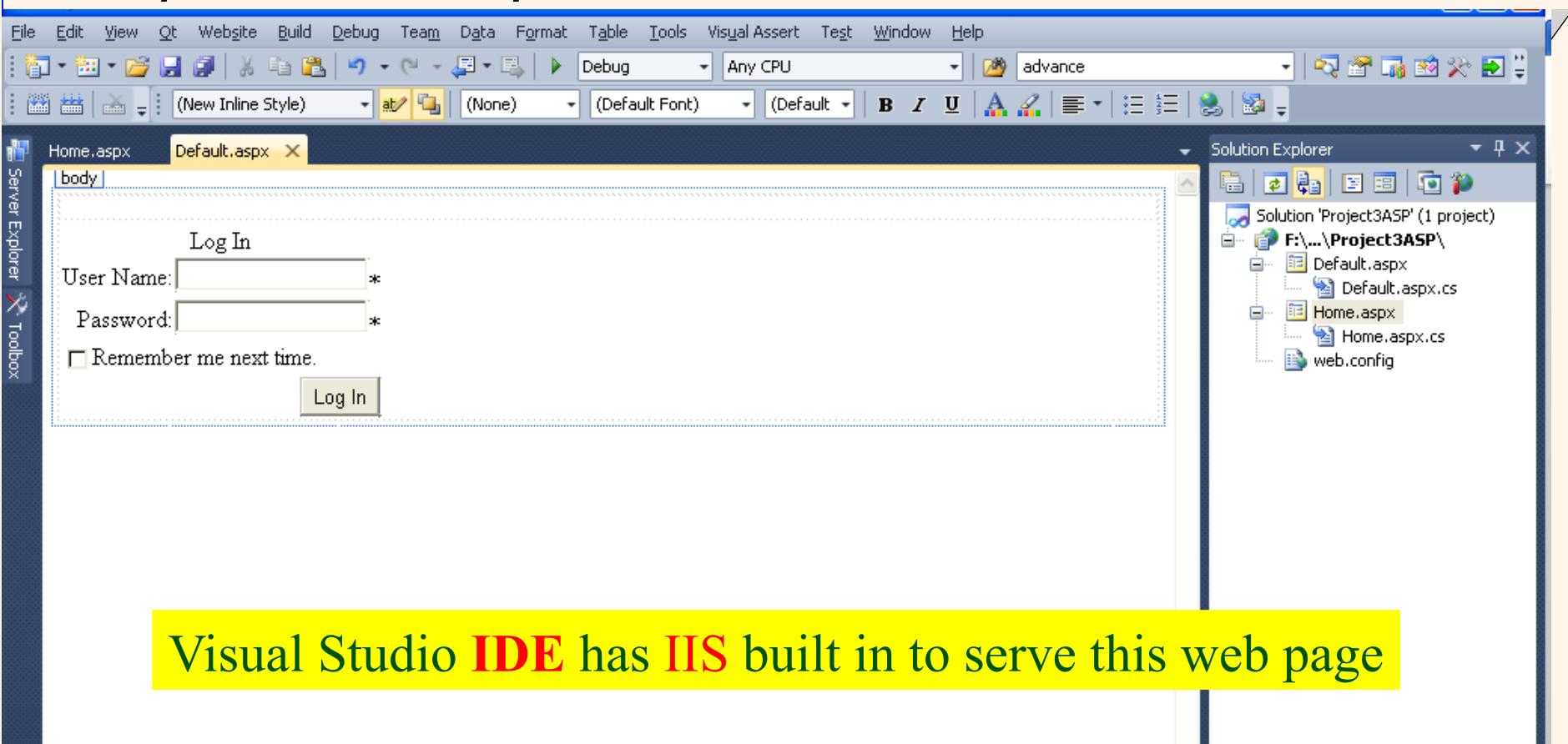
You Request an **ASP Page**
www.ocstc.org/default.aspx

ASP Compared to Other Languages



- ❖ **ASP** and /**PHP/Perl/CGI/JSP** execution are roughly equivalent in execution sequence
- ❖ Compared to **JavaScript**, don't have to worry about **Browser** versions or disabling
- ❖ Compared to **Java**, don't have to worry about whether JRE (or MVM) is installed

Complete Example



Visual Studio **IDE** has IIS built in to serve this web page

Source view provides a view of what is “under the hood” so to speak and displays the underlying HTML.

The design view is more of a WYSIWYG-type interface that greatly simplifies the design process

ASP to MS SQL DBMS Server : A Complete Example

The screenshot shows a Windows Internet Explorer window with the title bar "http://localhost:13202/Project3ASP/Default.aspx - Windows Internet Explorer provided by MSN & Bing". The address bar also displays "http://localhost:13202/Project3ASP/Default.aspx". The main content area contains a "Log In" form with fields for "User Name" (containing "vhilford") and "Password" (containing masked text). There is a "Remember me next time." checkbox and a "Log In" button.

Log In

User Name: vhilford

Password: ••••••••••

Remember me next time.

Log In

The screenshot shows a Windows Internet Explorer window with the title bar "http://localhost:13202/Project3ASP/Home.aspx - Windows Internet Explorer provided by MSN & Bing". The address bar displays "http://localhost:13202/Project3ASP/Home.aspx". The main content area displays the message "Welcome, you have SUCCESSFULLY logged in, vhilford !".

Welcome, you have SUCCESSFULLY logged in, vhilford !

ASP to MS SQL DBMS Server : A Complete Example

The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the Solution Explorer window displays a project named 'Project3ASP' with files: Default.aspx, Default.aspx.cs, Home.aspx, Home.aspx.cs, and web.config. The main workspace contains two tabs: 'Default.aspx.cs*' and 'Default.aspx'. The 'Default.aspx.cs*' tab shows C# code for the Default.aspx page. The code includes imports for System.Web.Security, System.Web.UI, System.Web.UI.WebControls, System.Web.UI.WebControls.WebParts, System.Web.UI.HtmlControls, and System.Data.SqlClient. It defines a partial class '_Default' that inherits from System.Web.UI.Page. The Page_Load event handler is present. The Login1Authenticate1 event handler contains logic to connect to a SQL Server database, retrieve user information, and set session variables if login is successful. The 'Default.aspx' tab shows the ASPX markup for the login form.

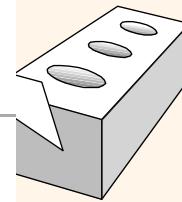
```
5  using System.Web.Security;
6  using System.Web.UI;
7  using System.Web.UI.WebControls;
8  using System.Web.UI.WebControls.WebParts;
9  using System.Web.UI.HtmlControls;
10 using System.Data.SqlClient;
11
12 public partial class _Default : System.Web.UI.Page
13 {
14     protected void Page_Load(object sender, EventArgs e)...
15
16     protected void Login1Authenticate1(object sender, AuthenticateEventArgs e)
17     {
18         SqlConnection con = new SqlConnection(@"Data Source=sqlserver.cs.uh.edu,1044;Initial Catalog=Project3");
19         SqlDataAdapter mysqlid = new SqlDataAdapter("SELECT UserID, Password FROM Login where User=" + Login1.UserName);
20         DataTable dt = new DataTable();
21
22         mysqlid.Fill(dt);
23         //If no rows returned
24         if (dt.Rows.Count == 0)
25         {
26             Login1.LoginButtonText = "Invalid ID or Password";
27             //Response.Redirect("Default.aspx");
28         }
29         else //User does exist and login successful
30         {
31             Session["id"] = Login1.UserName;
32             //Login is successful, redirect user to his home page
33             Response.Redirect("Home.aspx", true);
34         }
35     }
36
37     Session["id"] = Login1.UserName;
38 }
39
40 }
```

Default.aspx contains the user interface component of the application; the corresponding

Default.aspx.cs file contains any Visual C# programming code that may be included.

PHP

From Wikipedia, the free encyclopedia



This article is about the scripting language. For other uses, see [PHP \(disambiguation\)](#).

PHP is a general-purpose server-side scripting language originally designed for [Web development](#) to produce dynamic [Web pages](#). It is one of the first developed server-side scripting languages to be embedded into an [HTML](#) source document rather than calling an external file to process data. The code is [interpreted](#) by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a [command-line interface](#) capability and can be used in [standalone graphical applications](#).^[2] PHP can be deployed on most Web servers and also as a standalone [shell](#) on almost every [operating system](#) and [platform](#) free of charge.^[3] A competitor to Microsoft's [Active Server Pages](#) (ASP) server-side script engine^[4] and similar languages, PHP is installed on more than 20 million Web sites and 1 million [Web servers](#).^[5] Software that uses PHP includes [MediaWiki](#), [Joomla](#), [Wordpress](#), [Concrete5](#), [MyBB](#), and [Drupal](#).

PHP was originally created by [Rasmus Lerdorf](#) in 1995. The main implementation of PHP is now produced by [The PHP Group](#) and serves as the formal reference to the PHP language.^[6] PHP is free software released under the [PHP License](#), which is incompatible with the [GNU General Public License](#) (GPL) due to restrictions on the usage of the term *PHP*.^[7]

While PHP originally stood for *Personal Home Page*, it is now said to stand for *PHP: Hypertext Preprocessor*, a recursive acronym.^[8]

PHP to mySQL Server DBMS : A Complete Example

The screenshot shows a PHP development environment with the following details:

- Project Structure:** The left sidebar shows a project named "Project3PHP" containing "Source Files" and "index.php".
- File List:** The top bar lists "index.php" as the current file.
- Toolbar:** Standard file operations like Open, Save, Find, and Print are available.
- Code Editor:** The main area displays the PHP code for "index.php".

```
html head title
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2  <html>
3  |   <head>
4  |       <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5  |       <title>Project3PHP</title>
6  |   </head>
7  |   <body>
8  |       <?php
9  |           error_reporting(E_ALL); //turns on all error reporting, useful for debugging
10 |
11 |
12 |
13 |
14 |
15 |
16 |
17 |
18 |
19 |
20 |
21 |
22 |
23 |
24 |
25 |
26 |
27 |
28 |
29 |
30 |       class GuestbookDb {
|           function __construct($db, $user='root', $password='', $host='localhost') {
|               mysql_connect($host, $user, $password);
|               mysql_select_db($db);
|               //mysql_query('delete from address');
|           }
|
|           function addRecord($name, $note) {
|               $query = 'insert into guests values ("' . $name . '", "' . $note . '")'; //inserts
|               mysql_query($query);
|           }
|
|           function printRecords() {
|               $query = 'select * from guests'; //select all records
|               $result = mysql_query($query);
|
|               $guests = mysql_fetch_row($result); //gets array of a row of the results
|               while ($guests == true) {
|                   echo $guests[0] . '<br>' . $guests[1] . '<br><br>';
|               }
|           }
|       }
|   |</body>
| |</html>
```
- Annotations:** A yellow box with the word "PHP" is placed over the code editor area. A red rectangular box highlights the class definition and its methods.
- Status Bar:** The bottom right corner shows "5 | 27 INS".

PHP to MySQL DBMS Server : A Complete Example

The screenshot shows a PHP development environment with the following interface elements:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Project Explorer:** Shows "Project3PHP" with "Source Files" containing "index.php".
- Search Bar:** Search (Ctrl+I).
- Code Editor:** The "index.php" file is open. The code handles a guestbook database and prints records or adds new ones based on form submission. It includes a header redirection and an HTML form for user input.

Red Box Annotations:

- A red box highlights the line `$header="Location:index.php"`, which is part of a conditional block for handling form submissions.
- A red box highlights the entire `<form>` block, which contains the HTML form code.
- A blue box highlights the `value='Submit'` attribute of the `<input type="submit">` button.

Yellow Boxes:

- A yellow box labeled "action='index.php'" highlights the `action` attribute of the `<form>` tag.
- A yellow box labeled "HTML Form" highlights the entire `<form>` block.

```
36
37     if (isset($_POST['name']) & isset($_POST['note'])) //checks if form has been submitted
38         $guestBook = new GuestbookDb('guestbook');
39         $guestBook->addRecord($_POST['name'], $_POST['note']);
40         $guestBook->printRecords();
41         $header="Location:index.php" //file to redirect to. Will change depending on name of
42         header($header); //redirects browser to original page. By redirecting, we avoid multip
43         exit; //ends execution
44     }
45
46     $guestBook = new GuestbookDb('guestbook');
47     $guestBook->printRecords(); //Print records
48     ?>
49     <!-- Guestbook Form-->
50     <b>Please sign our guest book!</b><br>
51     <form action="" method="post" enctype="application/x-www-form-urlencoded">
52         Name: <input type="text" size="50" maxlength="100" name="name" /><br>
53         Note: <textarea rows="6" cols="40" name="note" ></textarea><br>
54
55         <br>
56         <input type="submit" value='Submit' />
57     </form>
58
59     </body>
60
61     </html>
62
```

TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE Django framework, PHP PhpStorm IDE Zend framework WEB APP, JAVA SPRING MVC WEB APP



Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



AJAX



ASP.NET MVC MSSQLSERVER WEBB APP



PYTHON PYCHARM DJANGO MYSQL WEB APP



PHP PYSTORM ZEND MYSQL WEB APP

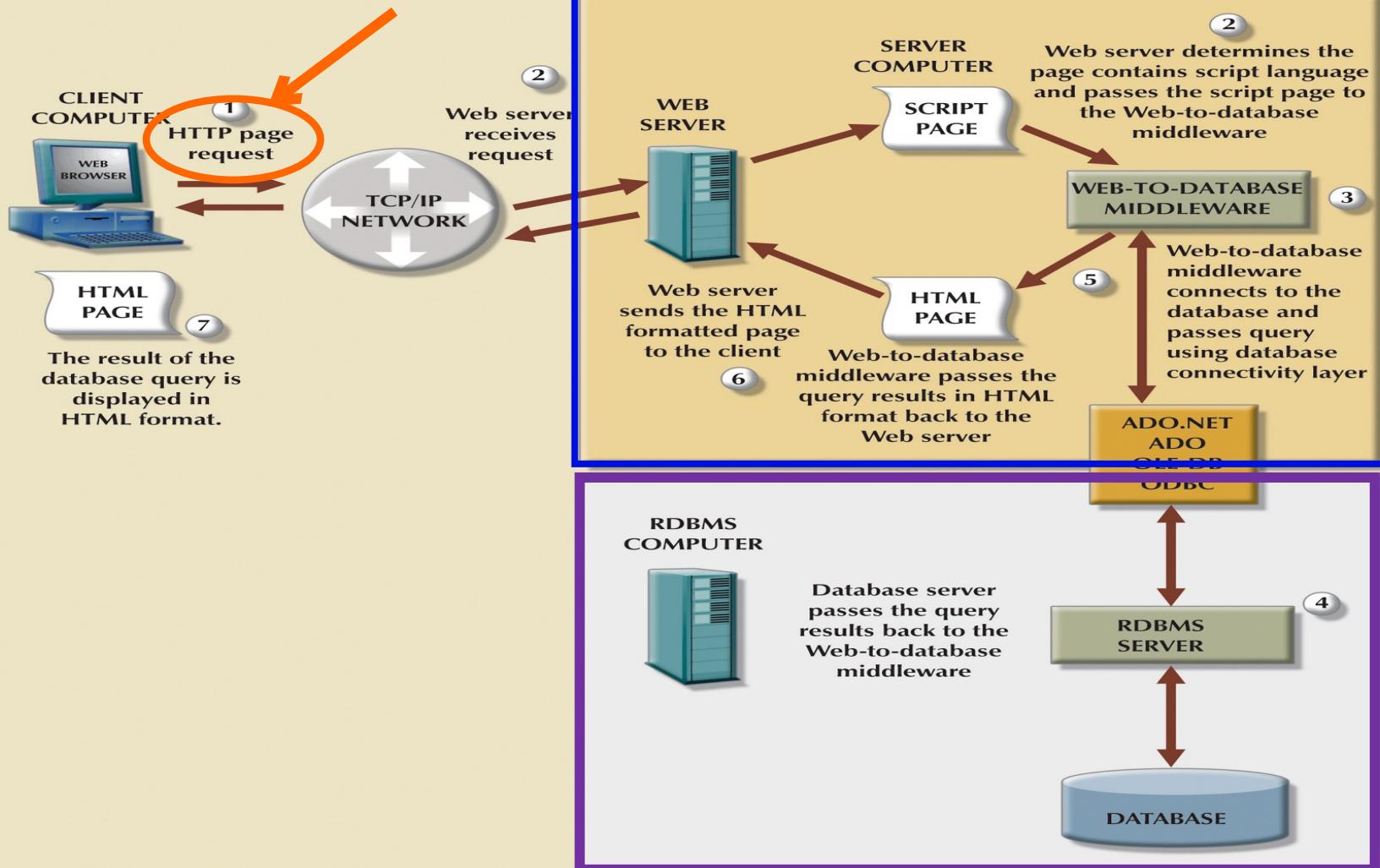


Ruby on Rails

Web Database Application Development

FIGURE
14.7

Web-to-database middleware



TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE Django framework, PHP PhpStorm IDE Zend framework WEBB APPS

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾

7.9 XMLHttpRequest (Ajax)

 Present

 Note

Ajax introduction

A normal **HTTP request** is triggered by clicking on a hyperlink or submitting a form, after which the browser may appear non-responsive while the browser waits for the server response. A page remains non-responsive until the request is sent to and processed by the server, and a **response** is returned to the browser. This delay may be undesirable for some web applications and may annoy users if the delay is long.

Ajax (Asynchronous JavaScript and XML) is a technique to asynchronously communicate with a server and update a web page once the response is received, without reloading the whole web page. An **asynchronous request** occurs when the web application sends a request to the server and continues running without waiting for the server response. Although the "x" in Ajax stands for "XML", Ajax is used to transmit plain text, HTML, XML, and JSON.

XMLHttpRequest is an object for communicating with web servers using Ajax. Using the XMLHttpRequest object allows web browsers to hide the communication latency and continue to provide a responsive user interface while waiting for a server response. The XMLHttpRequest object defines handlers for events that occur during the request/response cycle. Ex: A response arrives at the browser, an error occurs during a request, etc. Using event-driven programming, the web application can continue providing a responsive interface and does not need to wait for a response from the server. The web application later updates the page once the response is received



AJAX



AJAX Step by Step Tutorial

AJAX NetBeans JSP pages

AJAX Visual Studio ASP pages

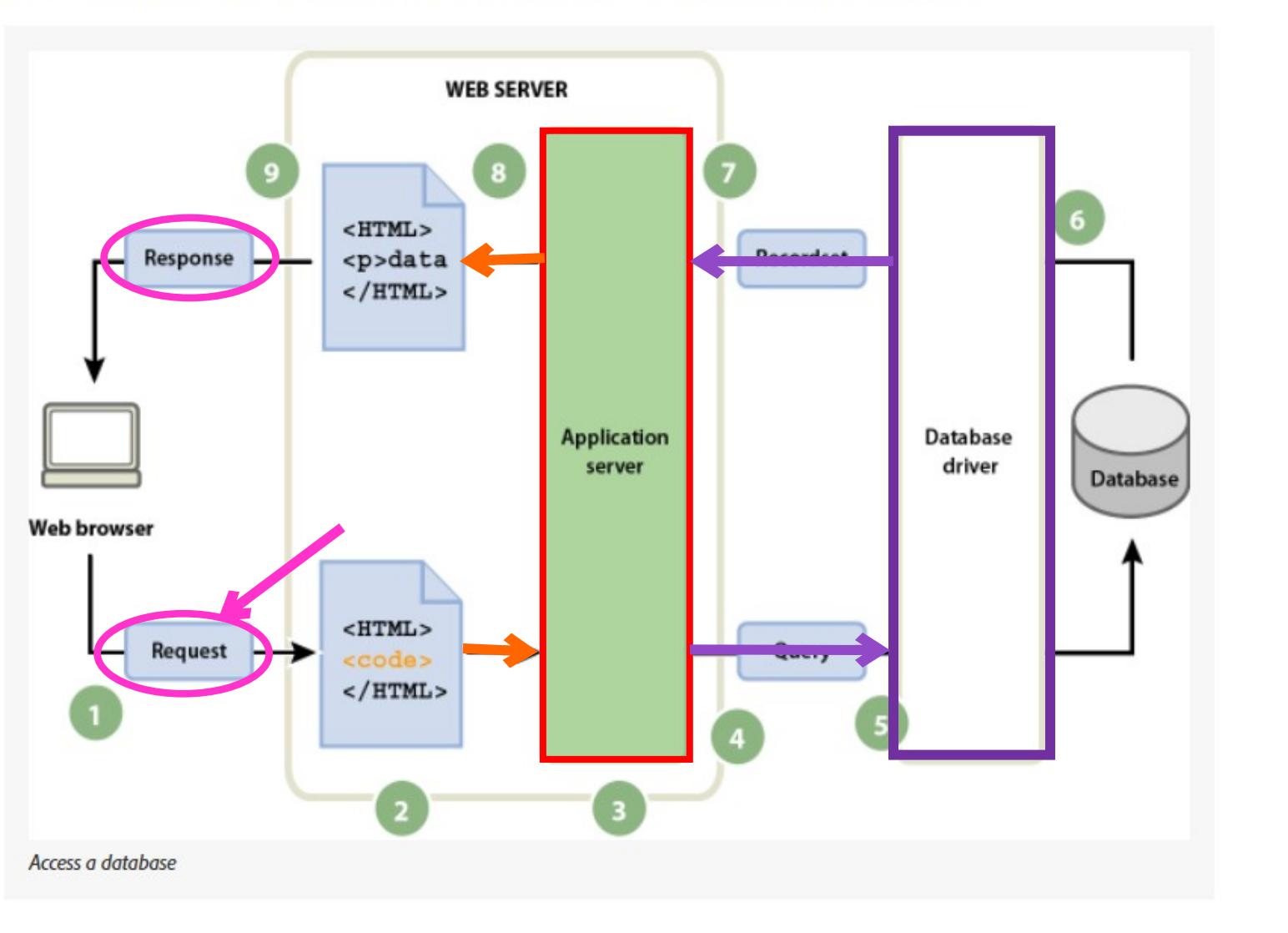


AJAX?

AJAX

Asynchronous JAVaScript & XML

Web **Database** Application Development



an error occurs during a request, etc. Using **event-driven programming**, the web application can continue providing a responsive interface and does not need to wait for a response from the server. The web application later updates the page once the response is received.

PARTICIPATION
ACTIVITY

7.9.1: Asynchronous HTTP request.

Start 2x speed

The text in the `<cite>` element usually renders in *italic*.

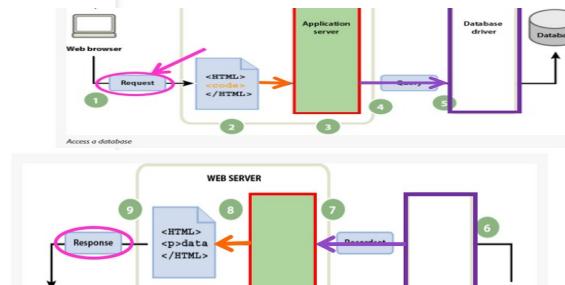
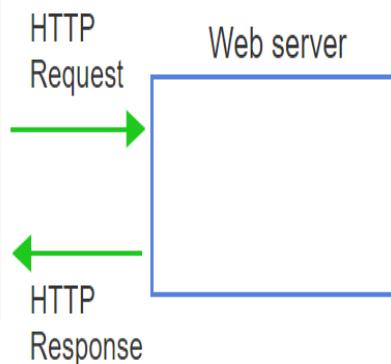
```
<h3>Movie Information</h3>
<div id="movieinfo">

    <!-- to be loaded -->

</div>
```

Movie Information

```
let movieinfo = document.getElementById("movieinfo");
let xhr = new XMLHttpRequest();
xhr.addEventListener("load", function() {
    movieinfo.innerHTML = xhr.response;
});
xhr.open("GET", "starwars.html");
xhr.send();
```





AJAX IDE's

- ▶ Visual Studio **C# ASP**



- ▶ Netbeans **JAVA JSP**



- ▶ Aptana



- ▶ Zend **PHP**

AJAX NetBeans JSP pages

Projects Files Services

MRT7java

- build
- empty
- web
- META-INF
- WEB-INF
- composer.jsp
- error.jsp
- index.jsp
- javascript.js
- stylesheet.css

nbproject

src

test

web

build.xml

File browser showing javascript.js code:

```
1 var req;
2 var isIE;
3 var completeField;
4 var completeTable;
5 var autoRow;
6
7 + function init() {...}
13
14 + function doCompletion() {...}
21
22 + function initRequest() {...}
33
34 + function callback() {...} red box
44
45 + function appendComposer(firstName,lastName,composerId) {...}
71
72 + function getElementY(element){...}
86
87 + function clearTable() {...}
95
96 + function parseMessages(responseXML) {...}
```

Diagram illustrating the AJAX architecture:

The diagram illustrates the three-layer architecture of an AJAX application:

- View:** Represented by a box containing a "Presentation" component.
- Model:** Represented by a box containing a "Data Store" component.
- Controller:** Represented by a box containing a "Business Logic Implementation (PHP, servlet, etc.)" component.

The flow of data is as follows:

- An "HTTP Request" is sent from the Browser to the Web Server.
- The Web Server processes the request and sends "XML Data" to the Business Logic Implementation.
- The Business Logic Implementation interacts with the Data Store.
- The Business Logic Implementation sends data back to the Web Server.
- The Web Server sends an "XMLHttpRequest" back to the Browser.
- The Browser's "callback()" function is triggered via a "JavaScript call".
- The "callback()" function receives "HTML & CSS data" and updates the "Presentation" component.

Auto-Completion using ... x

localhost:8084/MRT7java/

Auto-Completion using AJAX

This example shows how you can do real time auto-completion using Asynchronous JavaScript and XML (Ajax) interactions.

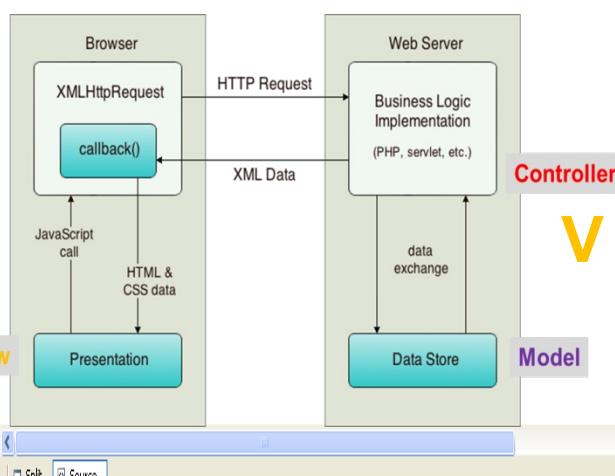
In the form below enter a name. Possible names that will be completed are displayed below the form. For example, try typing in "Bach," "Mozart," or "Stravinsky," then click on one of the selections to see composer details.

Composer Name: b

Bach

- Bedrich Smetana
- Benjamin Britten
- Leonard Bernstein
- Bela Bartok
- Johann Sebastian Bach
- Johannes Brahms
- Ludwig van Beethoven
- Georges Bizet
- Louis-Hector Berlioz

AJAX Visual Studio ASP pages



The diagram illustrates the MVC (Model-View-Controller) architecture. It shows three main components: View, Model, and Controller.

- View:** Represented by a box containing "Presentation". It interacts with the Model via a "JavaScript call" and receives "HTML & CSS data" from the Model.
- Model:** Represented by a box containing "Data Store". It interacts with the View via a "callback()" function in the View's XMLHttpRequest.
- Controller:** Represented by a box containing "Business Logic Implementation (PHP, servlet, etc.)". It receives "HTTP Request" from the View and sends "XML Data" to the View.

Annotations in the image:

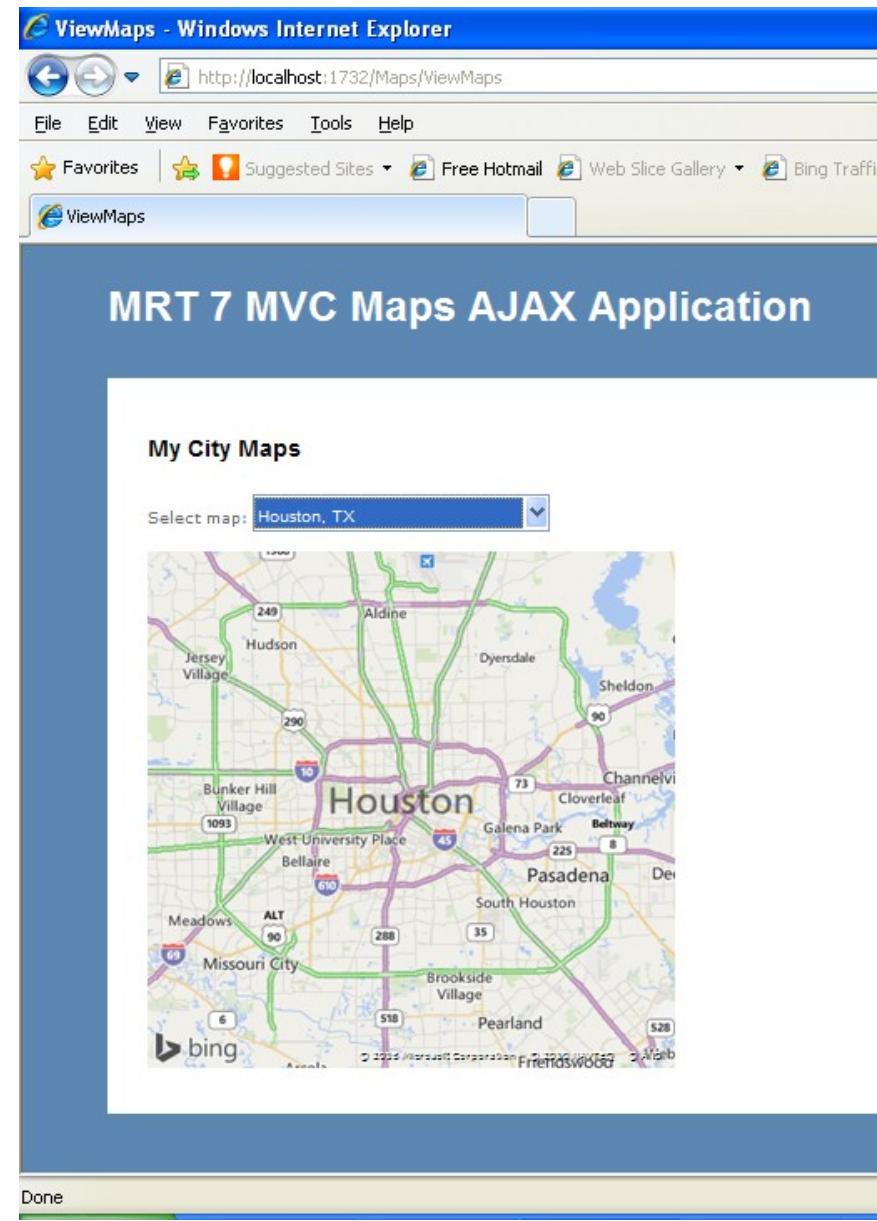
- C:** Points to the Controller box.
- M:** Points to the Model box.
- V:** Points to the View box.

Screenshot of Visual Studio showing the code for Index.aspx:

```
<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage" %>

<%><asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">
    Home Page
</asp:Content>

<%><asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <h2><%= Html.Encode(ViewData["Message"]) %</h2>
    <p>
        Page Rendered: <%= DateTime.Now.ToString() %<br />
        <br />
        <% using (Ajax.BeginForm("UpdateForm", new AjaxOptions{UpdateTargetId="textEntered"})) { %>
            <% Html.TextBox("textEntered", "Enter text")%>
            <input type="submit" value="Submit"/><br />
            <span id="textEntered">Nothing Entered</span>
        <% } %>
    </asp:Content>
```



**TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE
Django framework, PHP PhpStorm IDE Zend framework WEBB APPS**

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



ASP.NET MVC MSSQLSERVER WEBB APP

ASP.NET MVC MSSQLSERVER WEBB APP

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



Visual Studio 2015 ASP.NET MVC C# Step by Step Tutorial.docx

What's ASP.NET **MVC**

Microsoft **framework** that builds on top standard ASP.NET engine

ASP.NET **Web Form**

ASP.NET **MVC**

Core standard ASP.NET engine

Microsoft .NET Framework

Goals of ASP.NET MVC

- Testability
- Pluggable
- REST API for JSON data exchange*
- Existing ASP.NET feature still available
- Full control of HTML (HTML5)
- Open source

What is REST

REST is acronym for REpresentational State Transfer. It is architectural style for distributed hypermedia systems and was first presented by Roy Fielding in 2000 in his famous [dissertation](#).

Like any other architectural style, REST also does have its own [6 guiding constraints](#) which must be satisfied if an interface needs to be referred as RESTful. These principles are listed below.

Guiding Principles of REST

1. **Client-server** – By separating the user interface concerns from the data storage concerns, we improve the portability of the user interface across multiple platforms and improve scalability by simplifying the server components.
2. **Stateless** – Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client.
3. **Cacheable** – Cache constraints require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.
4. **Uniform interface** – By applying the software engineering principle of generality to the component interface, the overall system architecture is simplified and the visibility of interactions is improved. In order to obtain a uniform interface, multiple architectural constraints are needed to guide the behavior of components. REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state.
5. **Layered system** – The layered system style allows an architecture to be composed of hierarchical layers by constraining component behavior such that each component cannot "see" beyond the immediate layer with which they are interacting.
6. **Code on demand (optional)** – REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This simplifies clients by reducing the number of features required to be pre-implemented.

* JSON is an acronym for JavaScript Object Notation.

How does it work?

- **Controller** is responsible for handling incoming request

http://localhost:1725/**Hello**/**ActionTest**

Controller

Action in
controller

- **Routing system** decides how URLs map into particular **controllers** and **actions**

Routing

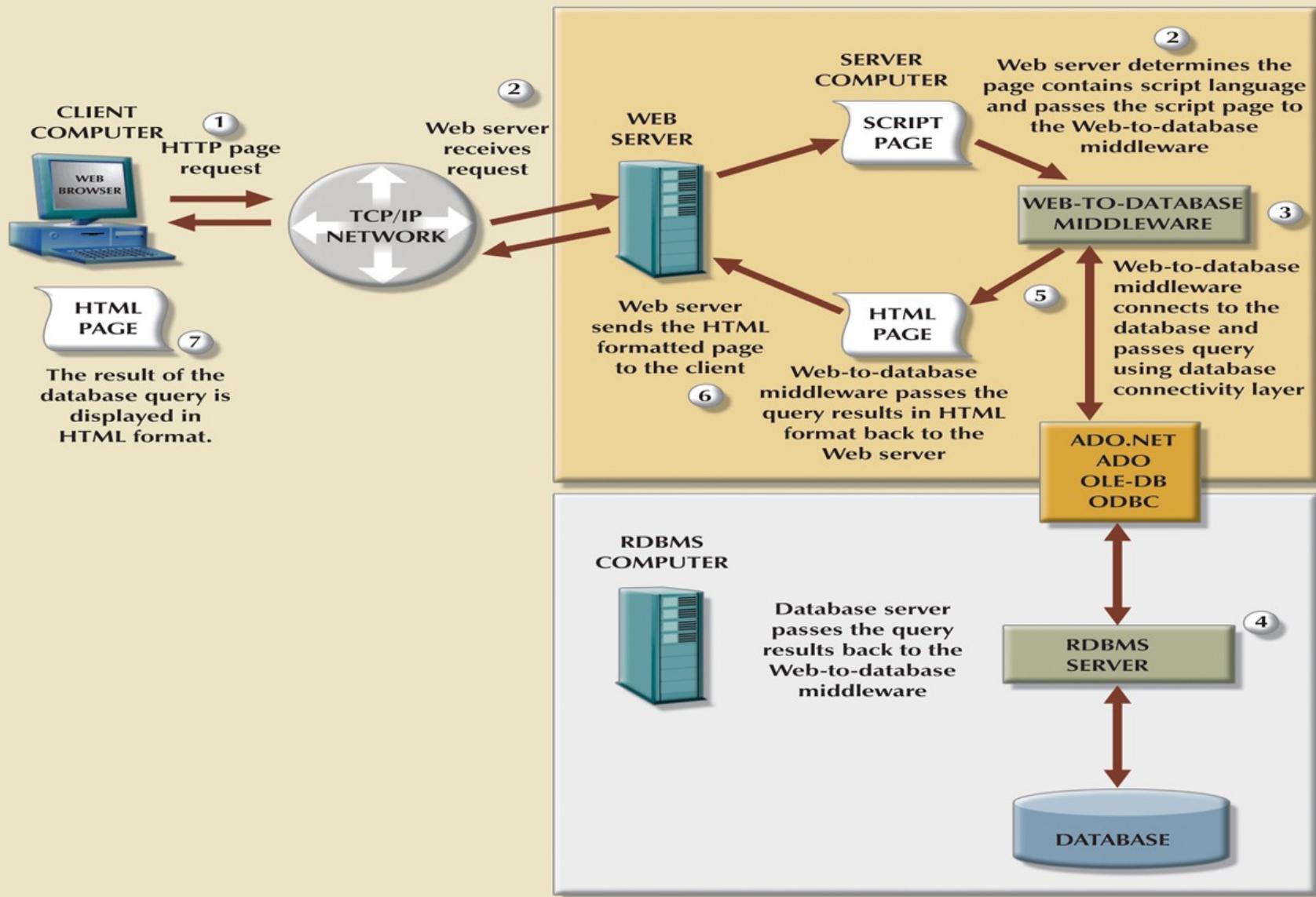
Component is first hit by a **request**
The rules are configured in **Global.asax.cs**

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.MapRoute(
        "Default",
        "{controller}/{action}/{id}",
        new { controller = "Home", action = "Index", id = "" });
}
```

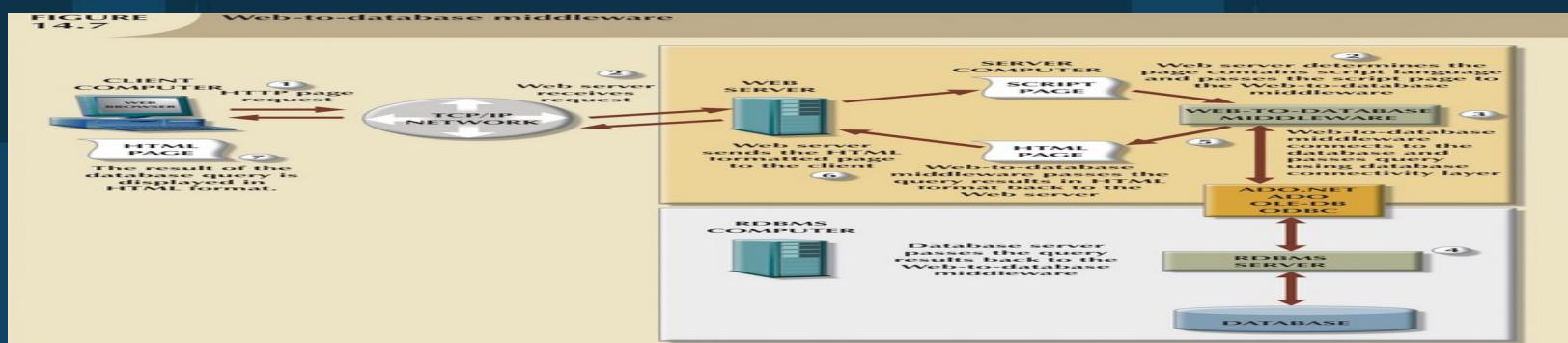
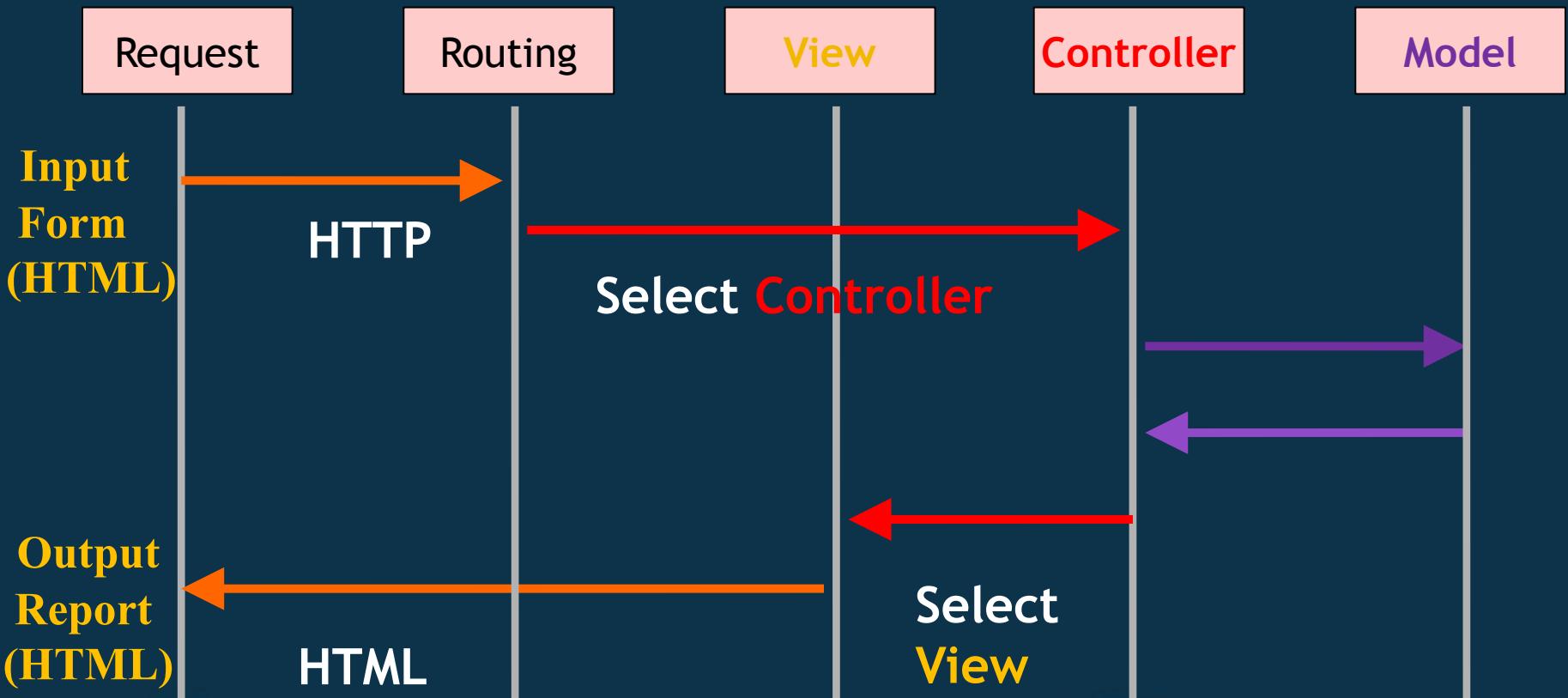
<http://localhost:1725/Product>Show/1>

FIGURE
14.7

Web-to-database middleware



Request Flow



TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE Django framework, PHP PhpStorm IDE Zend framework WEBB APPS

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾

PYTHON PYCHARM DJANGO MYSQL WEB APP

PYTHON PYCHARM DJANGO MYSQL WEB APP

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



PYTHON PyCharm Django Web App Step by Step Tutorial.docx

Download the **Python** Interpreter

- In your browser go to <https://www.python.org/download/releases/2.7.8/>
- Scroll Down to the Download Section
- Choose the version of python that matches your OS
- For windows choose the MSI installer and **not** the program database

Download

This is a production release. Please [report any bugs](#) you encounter.

We currently support these formats for download:

- [Windows x86 MSI Installer \(2.7.8\)](#)
- [Windows x86 MSI program database \(2.7.8\)](#)
- [Windows X86-64 MSI Installer \(2.7.8\) \[1\]](#)
- [Windows X86-64 MSI program database \(2.7.8\) \[1\]](#)
- [Windows help file](#)
- [Mac OS X 64-bit/32-bit x86-64/i386 Installer \(2.7.8\) for Mac OS X 10.6 and later \[2\] \(sig\).](#) [You may need an updated Tcl/Tk install to run IDLE or use Tkinter, see note 2 for instructions.]
- [Mac OS X 32-bit i386/PPC Installer \(2.7.8\) for Mac OS X 10.5 and later \[2\] \(sig\).](#)
- [Mac OS X 32-bit i386/PPC Installer \(2.7.8\) for Mac OS X 10.3 and later \(deprecated, see below\) \[2\] \(sig\).](#)
- [XZ compressed source tar ball \(2.7.8\) \(sig\)](#)
- [Gzipped source tar ball \(2.7.8\) \(sig\)](#)

Install **Python** Interpreter

- Click **next on all prompts** until the software is installed

Download PyCharm IDE Community Edition

- In your browser go to
<http://www.jetbrains.com/pycharm/download/>
- Click the Download Community button on the right

Windows Mac OS X Linux See what's new in PyCharm 3.4 »

Version: 3.4.1 Build: 135.1057 Released: June 10, 2014 [System requirements](#) [Installation Instructions](#)

 Professional Edition [Free 30-day trial](#)

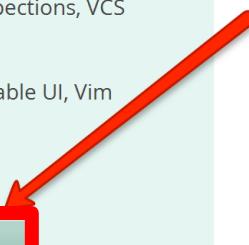
- Full-featured IDE for Python & Web development
- Supports Django, Flask, Google App Engine, Pyramid, web2py
- JavaScript, CoffeeScript, TypeScript, CSS, Cython, Template languages and more
- Remote development, Databases and SQL support, UML & SQLAlchemy Diagrams

 [Download Professional](#)

Community Edition [FREE](#)

- Lightweight IDE for Python development only
- Free, open-source, [Apache 2 license](#)
- Intelligent Editor, Debugger, Refactorings, Inspections, VCS integration
- Project Navigation, Testing support, Customizable UI, Vim key bindings

 [Download Community](#)



Install PyCharm IDE Community Edition

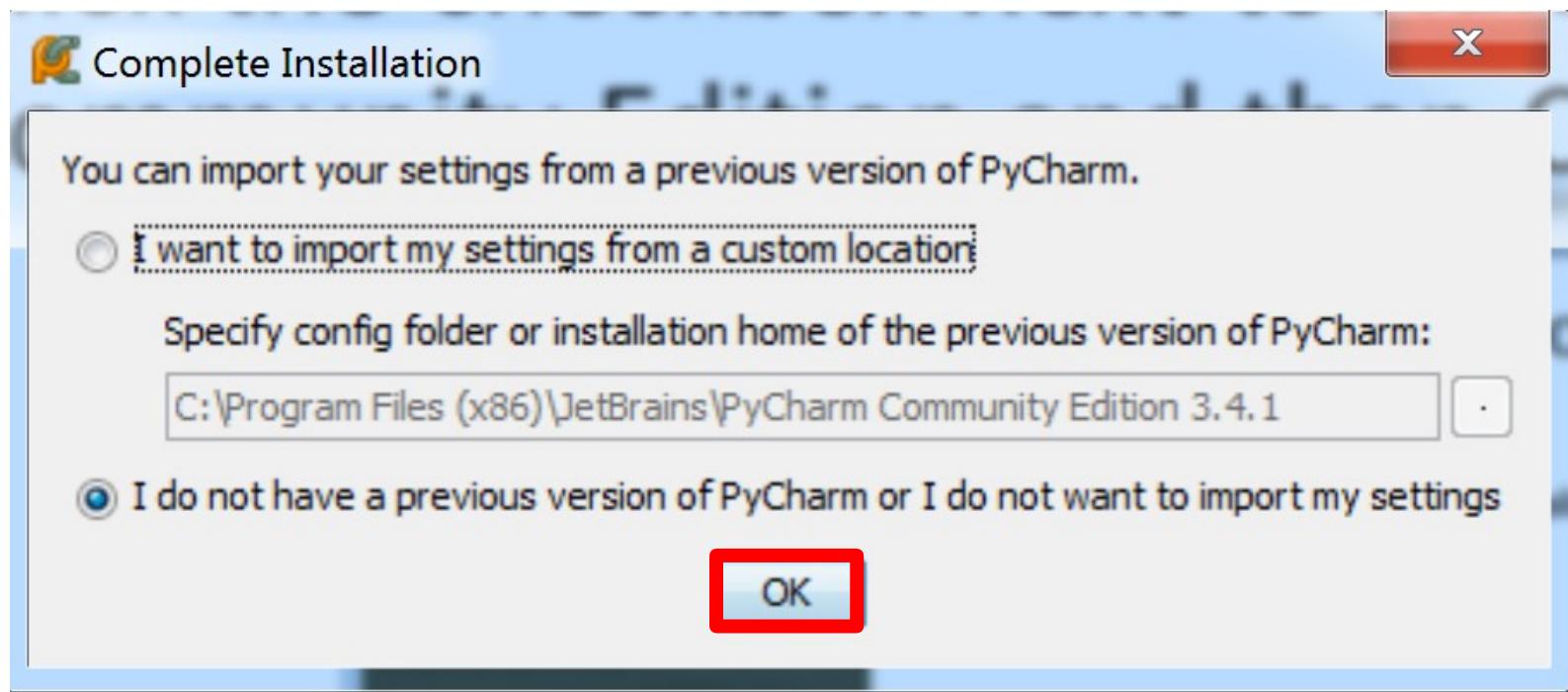
Click **Next** on all prompts until you get to the following screen

Click the checkbox next to Run Pycharm Community Edition and then Click **Finish**



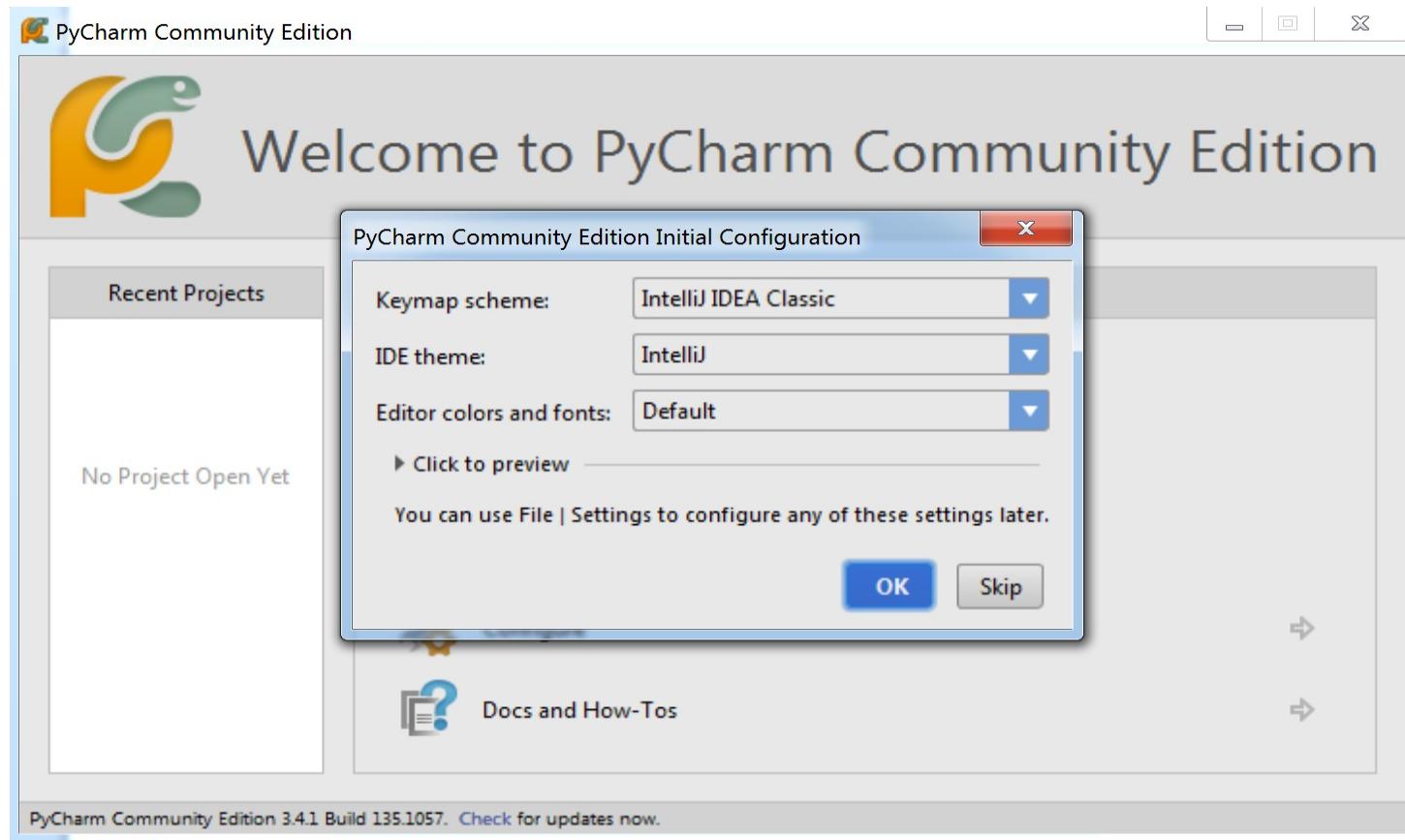
Install PyCharm IDE Community Edition

Make sure the “I do not have a previous version of PyCharm or I do not want to import my settings” is selected and click the OK button



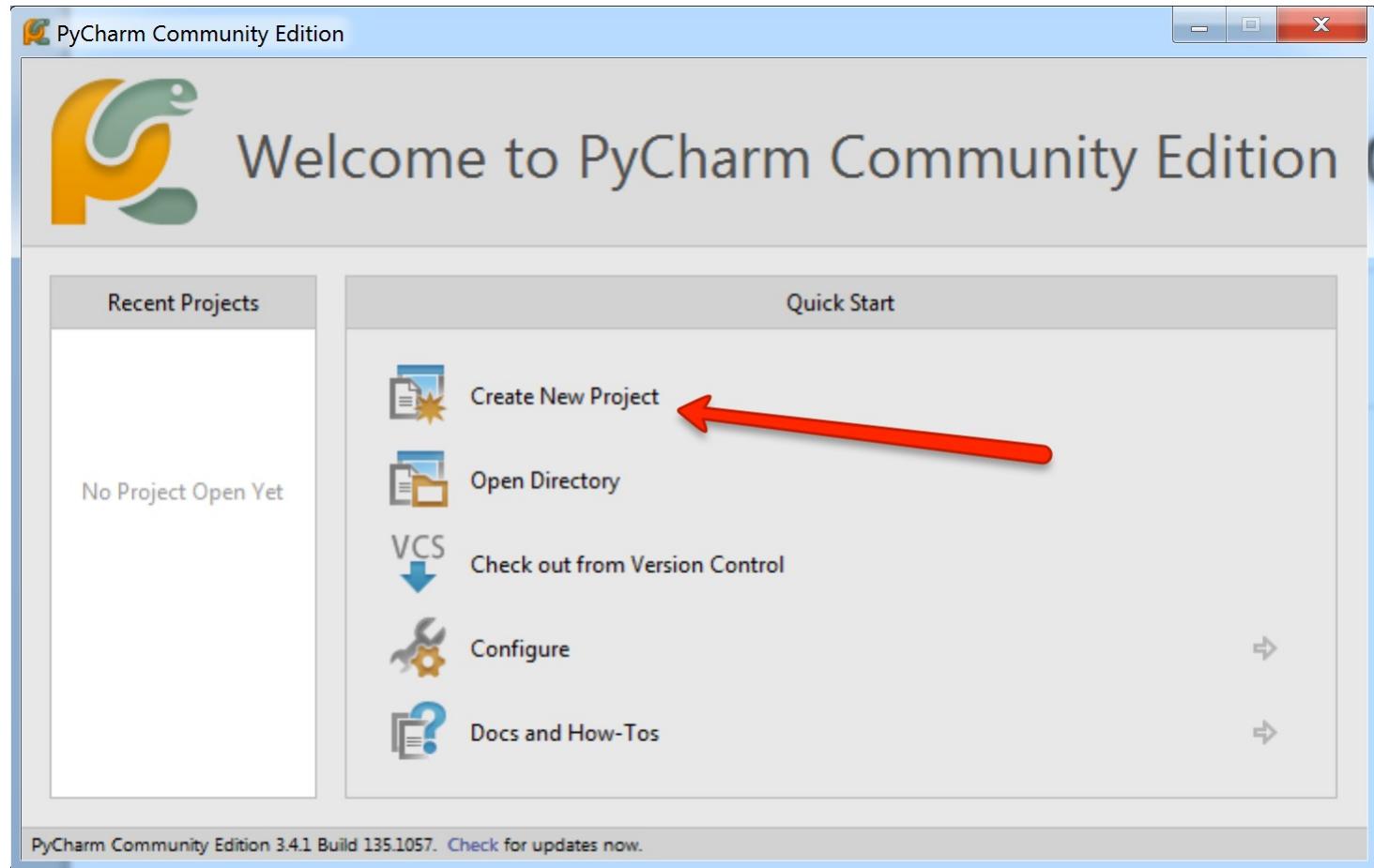
Install PyCharm IDE Community Edition

- Click the **OK** button



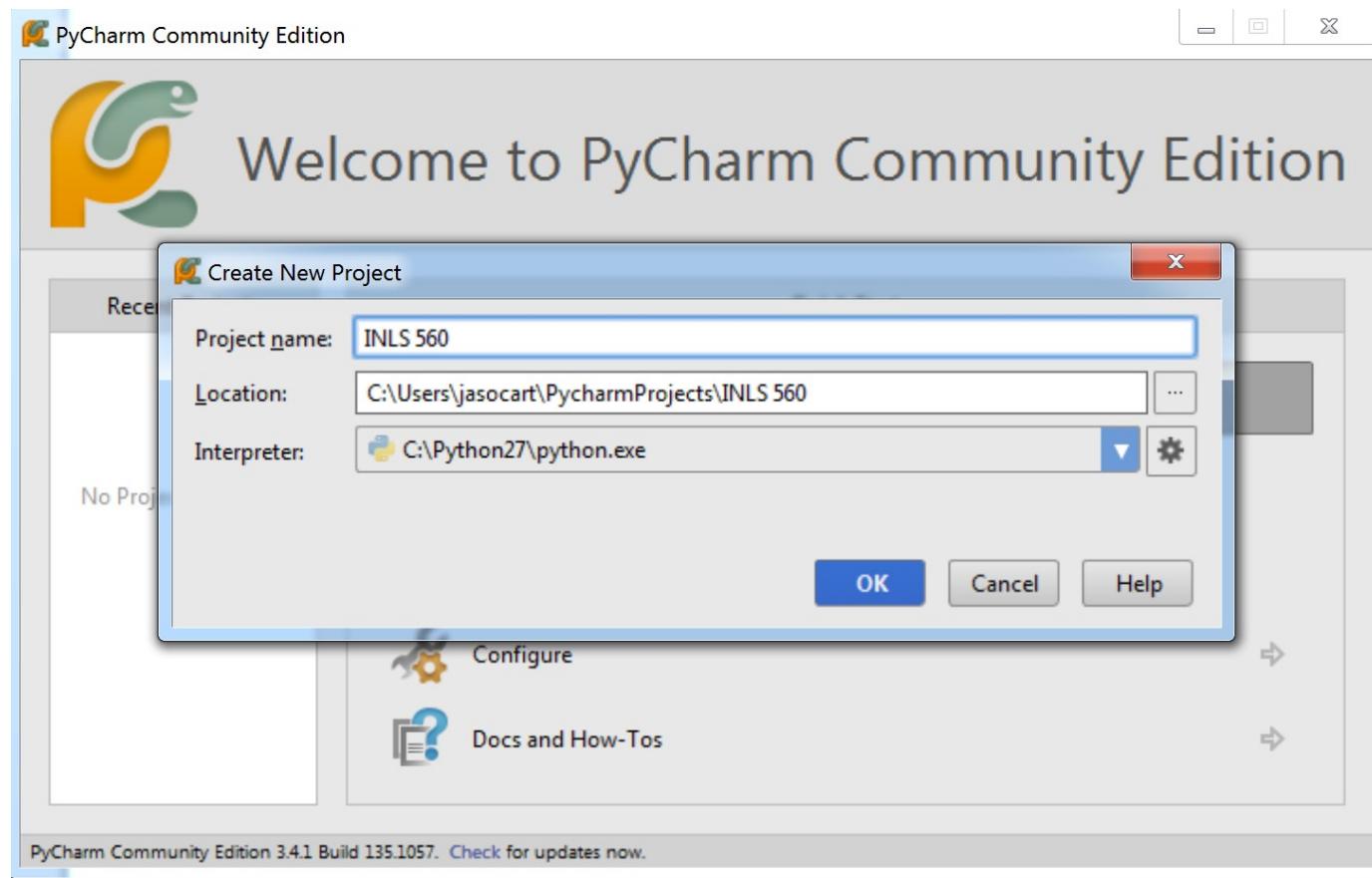
As easy as installing NetBeans, Eclipse, etc. **IDEs**

Create a new Project

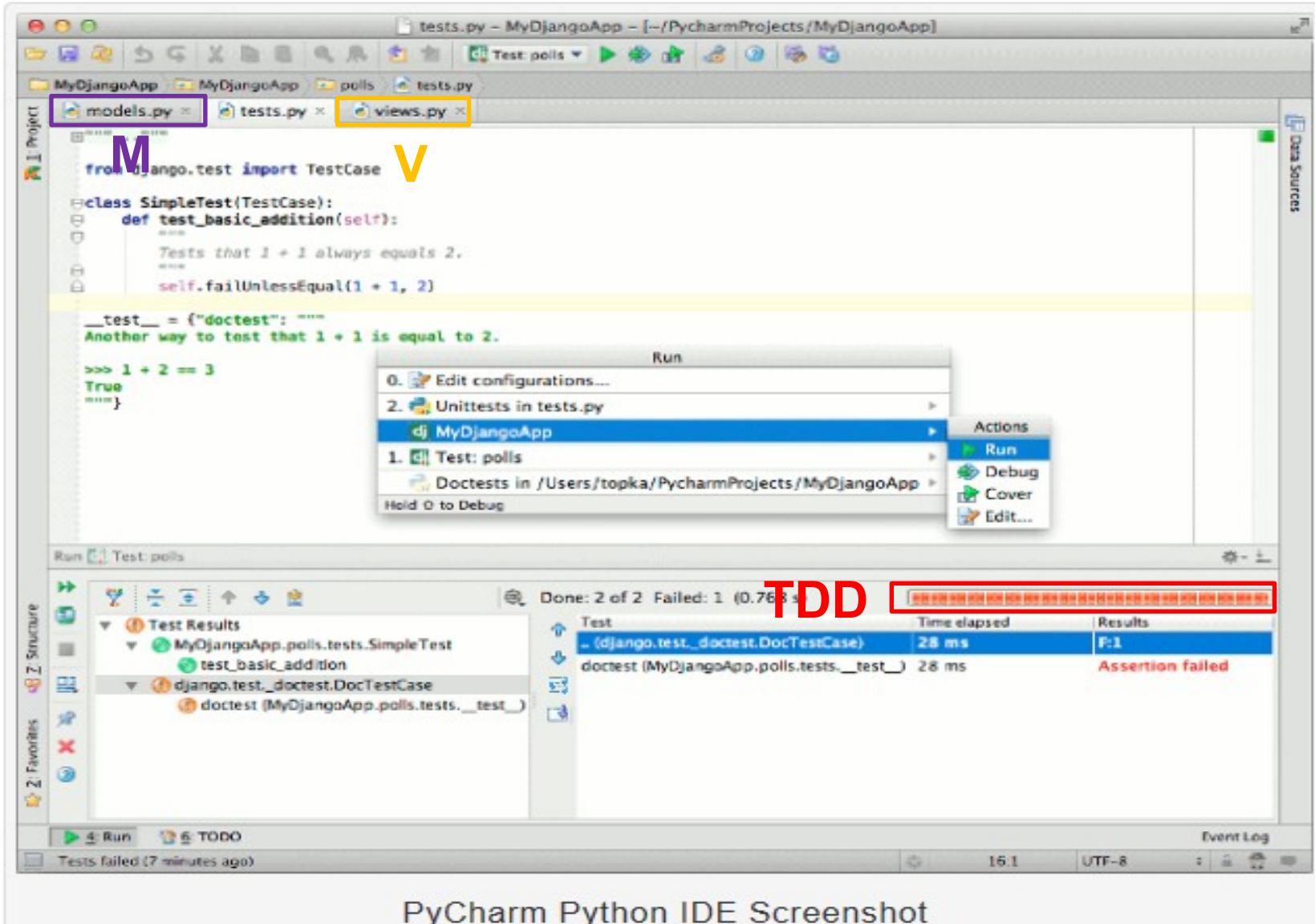


Create a new Project

- In the Project Name textbox type a project name, in this screenshot the project name is **INLS 560** and click the **OK** button



PyCharm IDE with Django MVC Framework



PyCharm Python IDE Screenshot

Using an MVC framework makes building web applications easy

TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE Django framework, PHP PhpStorm IDE Zend framework WEBB APPS

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



PHP PYSTORM ZEND MYSQL WEB APP



PHP ZEND 1.11.1 MYSQL WEB APP Step by Step Tutorial



PHP ZEND 2.0 MYSQL WEB APP Step by Step Tutorial

Downloads:

1. Download / Install **WAMP**

<http://www.wampserver.com/en/download.php>

<http://www.simplehelp.net/2008/08/25/how-to-install-and-setup-apache-mysql-and-php-in-windows/>

2. Make sure **WAMP** server is running (**green**)



WAMP Server runs on port **88**

<http://localhost:88>

ZendPHP For Windows Now Available!

ⓘ Some content in this message has been blocked because the sender isn't in your Safe senders list. I trust content from info@perforce.com. | Show blocked content

ZP Zend by Perforce <info@perforce.com>
Tue 2/1/2022 12:05 PM
To: vhilford@cs.uh.edu



Hi Victoria,

We are pleased to announce the immediate availability of ZendPHP for Windows!

ZendPHP for Windows marks the first commercial distribution of PHP for the Windows platform and covers Windows 7, 8, 10, 11, and Windows Server 2019.

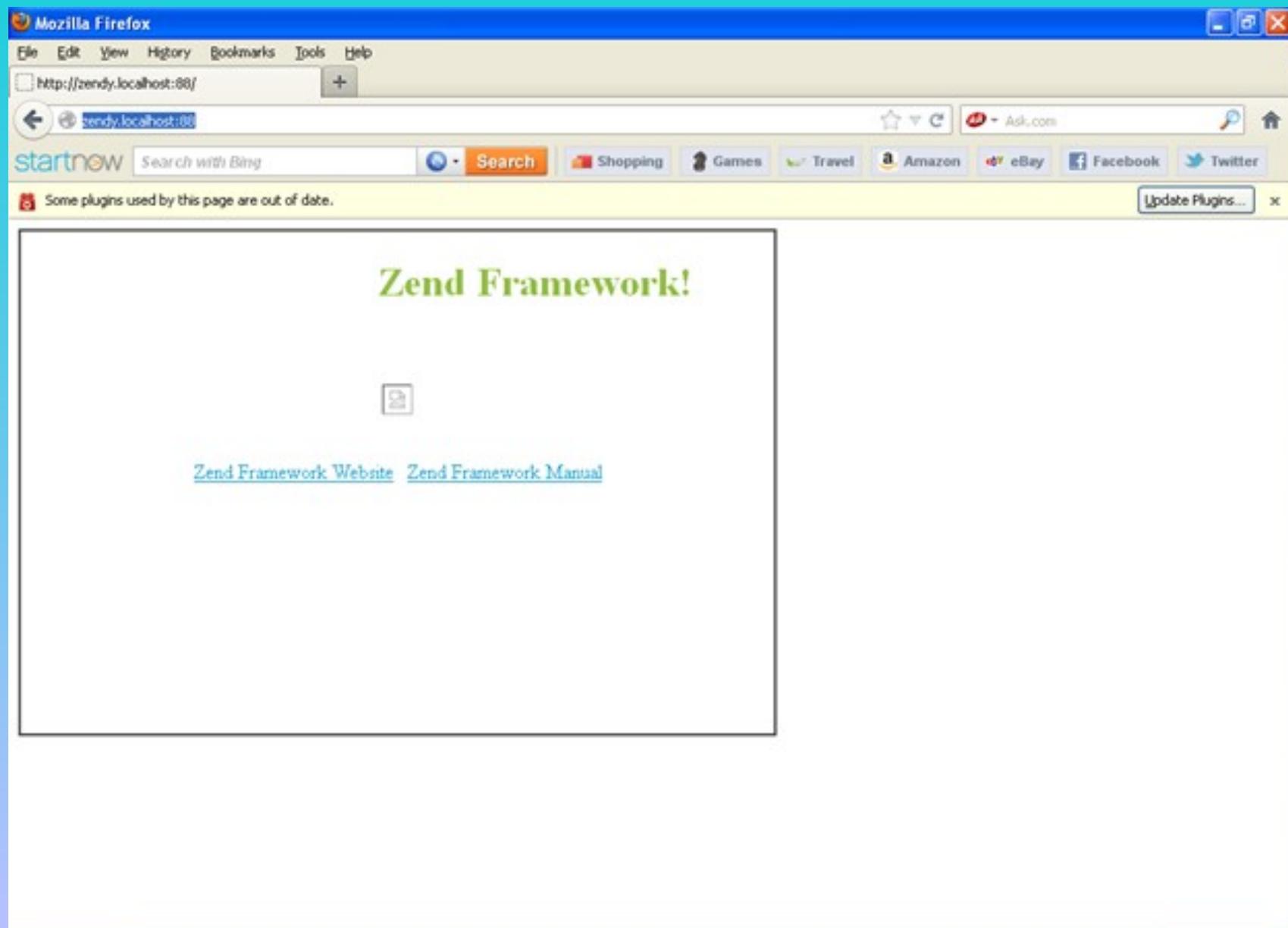
We are also providing the first Long-Term Support editions of PHP for Windows with versions 7.1, 7.2, and 7.3 available to customers, as well as versions under community support (currently PHP 7.4, 8.0, and 8.1).

To view the full release details and download ZendPHP for Windows, visit our [What's New page](#).

Thank you for your support,

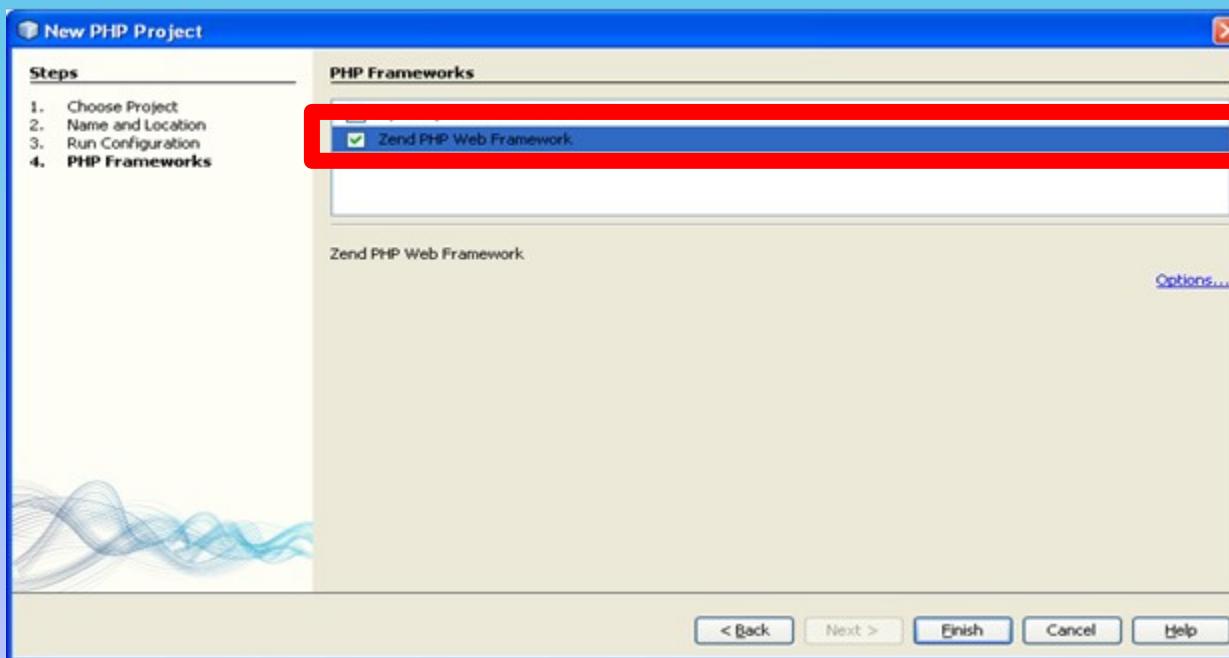
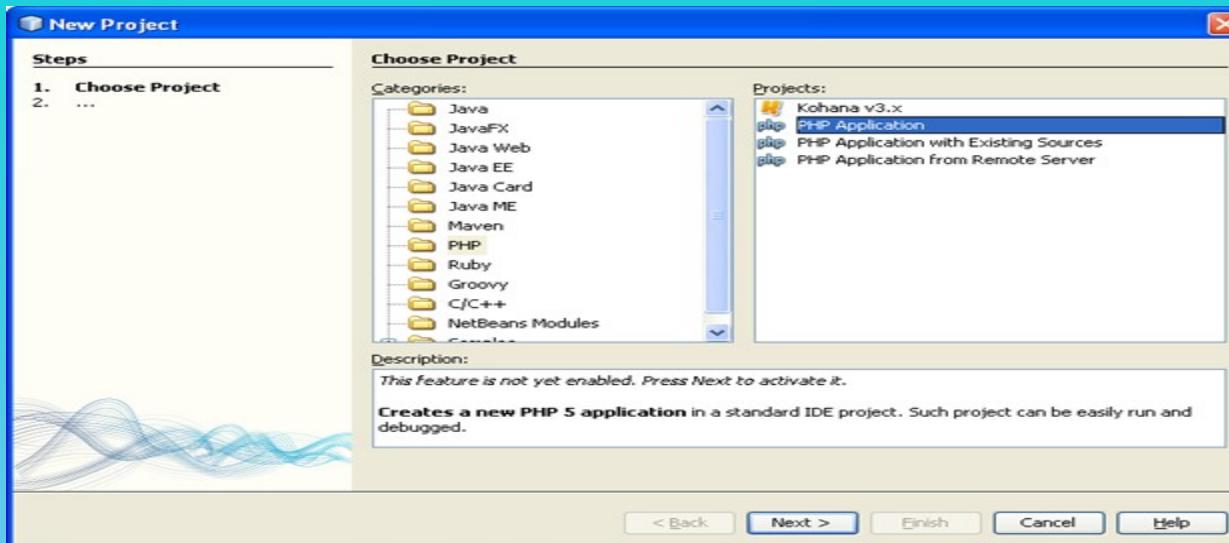
The Zend by Perforce Team

PHP Zend Framework Tutorial



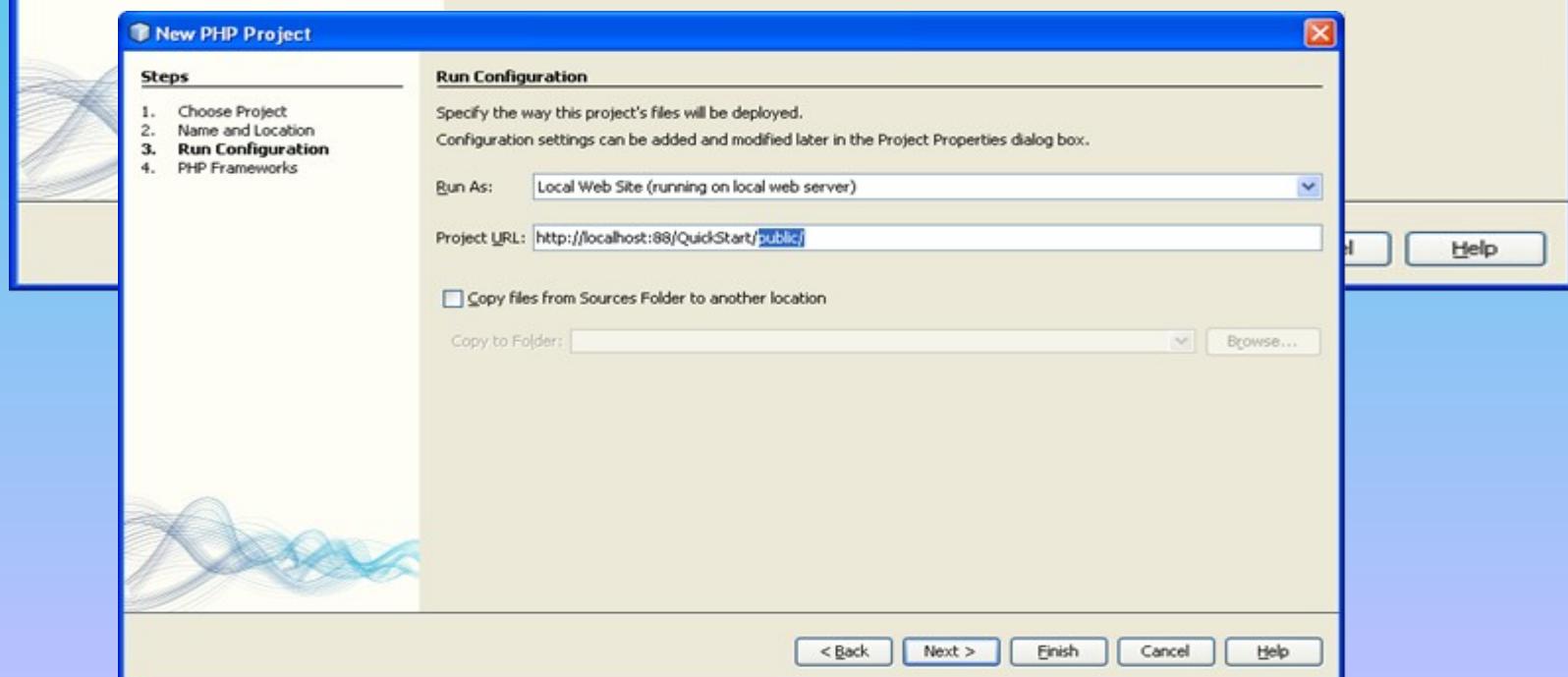
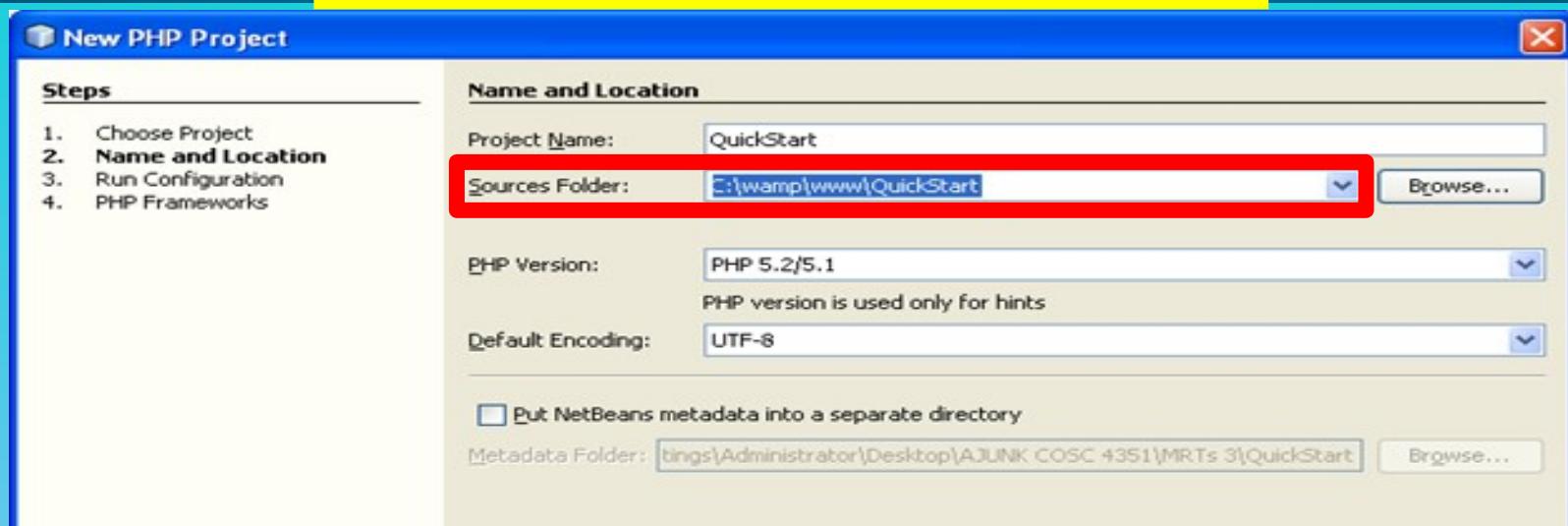
PHP Zend Framework Tutorial

NetBeans PHP Zend Framework Project



PHP Zend Framework Tutorial

NetBeans PHP Zend Framework Project Wamp



PHP Zend Framework Tutorial

NetBeans PHP Zend Framework Project Wamp

QuickStart - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

default

Search (Ctrl+F)

Project Files Services

QuidStart

Source Files

application

configs application.ini

controllers ErrorController.php IndexController.php

models

views Bootstrap.php

docs library public index.php

Test Files Include Path

C
M
V

```
<?php

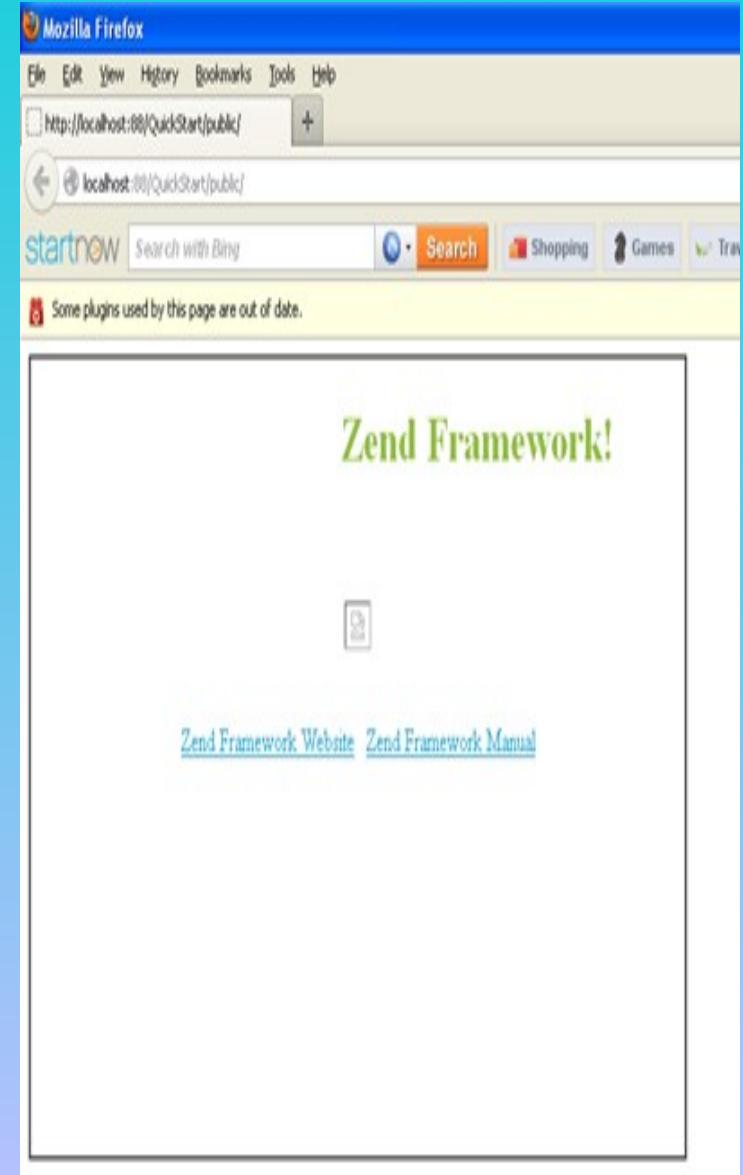
// Define path to application directory
defined('APPLICATION_PATH')
    || define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/../application'));

// Define application environment
defined('APPLICATION_ENV')
    || define('APPLICATION_ENV', (getenv('APPLICATION_ENV') ? getenv('APPLICATION_ENV')
        : 'development'));

// Ensure library/ is on include_path
set_include_path(implode(PATH_SEPARATOR, array(
    realpath(APPLICATION_PATH . '/../library'),
    get_include_path(),
)));

/** Zend_Application */
require_once 'Zend/Application.php';

// Create application, bootstrap, and run
$application = new Zend_Application(
    APPLICATION_ENV,
    APPLICATION_PATH . '/configs/application.ini'
);
$application->bootstrap()
    ->run();
```



ZF Zend **Framework**

- **ZF's flexible Model-View-Controller (MVC) implementation**
 - Keeps your code organized logically and handles “plumbing”
- **Components (Libraries)**
 - Including **Classes** to access web services

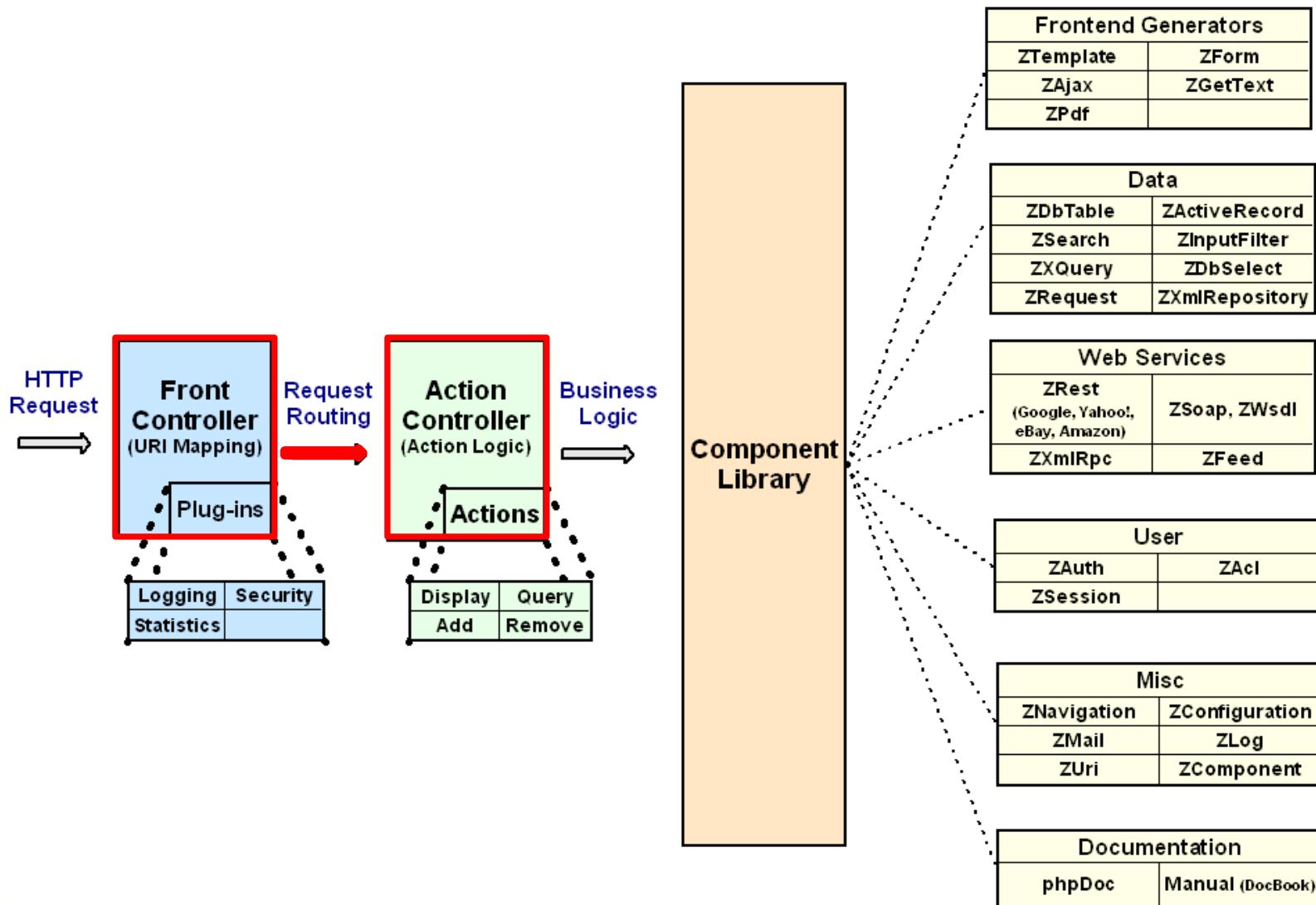
Model – View – Controller **Design Pattern**

- **Model**
 - Classes that access your data
- **View**
 - Templates to present that data (e.g. **HTML** for browser)
- **Controller (action controller)**
 - **Application** flow
 - Connects **Model** and **View**
- (Bonus: front Controller!)

Front Controller Design Pattern

- Front Controller sits in front of MVC
- All PHP requests funneled through index.php (bootstrap file)
- Front Controller gets your application started
 - Initializes request/response objects
 - Can handle common settings and functionality
 - “Include” paths
 - Configurations
 - Location of MVC components (if necessary)
 - Logging, db (perhaps), authentication/authorization
 - Converts URL to a “request” object with distinct parts
 - Routes requests to appropriate action Controllers
- Receives exceptions

Front Controller to action Controller



Controller example

The screenshot shows a PHP development environment with the following structure:

- PHP Explorer:** A tree view of the project structure under "demo".
 - application:** Contains "default", "controller", "helpers", "layouts", "models", "views", "bootstrap.php", "Initializer.php", "library", "public", "test", "Include Paths", and "JavaScript Support".
 - controller:** Contains "ErrorController.php" and "IndexController.php". The "IndexController.php" file is currently selected.
 - views:** Contains "V" (View).
- IndexController.php:** The code editor window displays the following PHP code:

```
1<?php
2
3require_once 'Zend/Controller/Action.php';
4
5class IndexController extends Zend_Controller_Action
6{
7    /**
8     * The default action - show the home page
9     */
10    public function indexAction()
11    {
12        // Use default value of 1 if id is not set
13        $id = $this->_getParam('id', 1);
14
15        // assign id to view
16        $this->view->id = $id;
17    }
18}
```

A yellow box highlights the line `$this->view->id = $id;`. A large purple box surrounds the entire code block.

View

- **Scripts (templates)**
 - PHP-based script templates to present data
 - Should contain only **display logic**, not **business logic**
 - Default naming: “myaction.**phtml**”
- **Helpers**
 - **Classes** and **methods** that provide reusable **View** functionality
 - Examples of built-in view helpers: escape(), formText(), partial(), partialLoop(), headTitle()
 - Write your own, too
- **Output filters**
- **Layout**
- **Placeholders**

Controller...leads to View

The screenshot shows a PHP development environment with the following structure:

- PHP Explorer:** A tree view of the project structure under "demo".
 - application:** Contains "default", "controller", "helpers", "layouts", "models", and "views".
 - controller:** Contains "ErrorController.php" and "IndexController.php". The "controller" folder is highlighted with a red box, and a large red letter "C" is overlaid on it.
 - bootstrap.php**
 - Initializer.php**
 - library**
 - public**
 - test**
 - Include Paths**
 - JavaScript Support**
- IndexController.php:** The code editor window displays the following PHP code:

```
1<?php
2
3require_once 'Zend/Controller/Action.php';
4
5class IndexController extends Zend_Controller_Action
6{
7    /**
8     * The default action - show the home page
9     */
10    public function indexAction()
11    {
12        // Use default value of 1 if id is not set
13        $id = $this->_getParam('id', 1);
14
15        // assign id to view
16        $this->view->id = $id;
17    }
18}
```

A yellow rectangular box highlights the assignment statement `$this->view->id = $id;`.

View script automatically rendered

The screenshot illustrates the Zend Framework development environment in Zend Studio. On the left, the PHP Explorer shows the project structure under 'demo': application (controllers, default, helpers, layouts, models, views), and bootstrap.php. The views folder contains filters, helpers, scripts, error, and index. Inside index, index.phtml is selected. The main window displays the code for IndexController.php:

```
1<?php
2
3/**
4 * Home page view
5 *
6 * @author Alan Seiden <alan@alanseniden.com>
7 * @version 1.00
8 */
9
10$this->headTitle('Zend Framework Demo');
11$this->placeholder('title')->set('Welcome');
12?>
13
14Welcome to the home page's simple view script.<BR>
15Your id is <?php echo $this->escape($this->id) ?>.
16
```

A red box highlights the placeholder code at line 15. A yellow arrow points from the left towards the rendered output. The bottom section shows two browser panes. The left pane shows the URL <http://localhost/zfdemo/> and displays the rendered output: "Welcome" followed by "Welcome to the home page's simple view script. Your id is 1.". The right pane shows the URL <http://localhost/zfdemo/index/index/id/39> and displays the rendered output: "Welcome" followed by "Welcome to the home page's simple view script. Your id is 39".

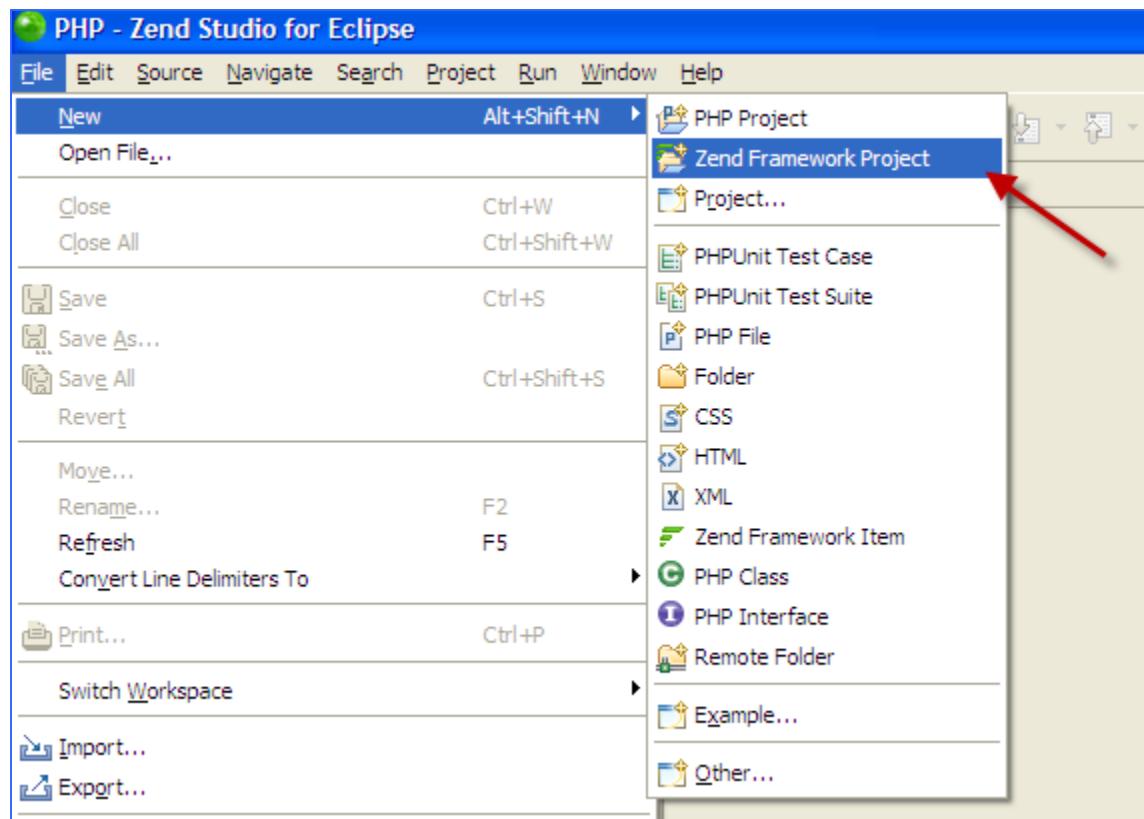
Zend_Layout / Master Page

The screenshot shows the Eclipse IDE's PHP Explorer interface. On the left, the file structure of a 'demo' application is displayed, including 'application', 'default', 'controllers', 'helpers', 'layouts', 'models', and 'views' directories. Inside 'views', there are 'filters', 'helpers', and 'scripts' sub-directories, with 'index.phtml' being the selected file. The main editor window on the right contains the PHP code for 'main.phtml'. Several sections of the code are highlighted with red boxes:

```
1<?php
2
3echo 'xml version="1.0" encoding="UTF-8" ?';
4echo $this->doctype()
5?>
6
7<html>
8
9    <head>
10        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
11        <?php
12            echo $this->headTitle();
13            echo $this->headScript();
14            echo $this->headStyle();
15        ?>
16    </head>
17
18    <body>
19        <h1><?php echo $this->placeholder('title') ?></h1>
20        <?php echo $this->layout()->content ?>
21
22        <br />
23        <br />
24    </body>
25
26</html>
```

Tools can help get you started

- Zend Studio for Eclipse creates default directory structures and Classes for you
 - Free trial: <http://www zend com/en/products/studio/>



TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE
Django framework, PHP PhpStorm IDE Zend framework WEBB APPS

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



Ruby on Rails

Ruby on Rails

Build Content ▾

Assessments ▾

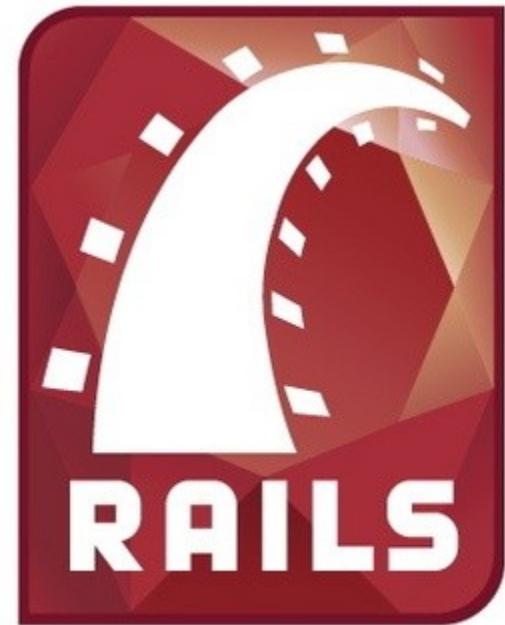
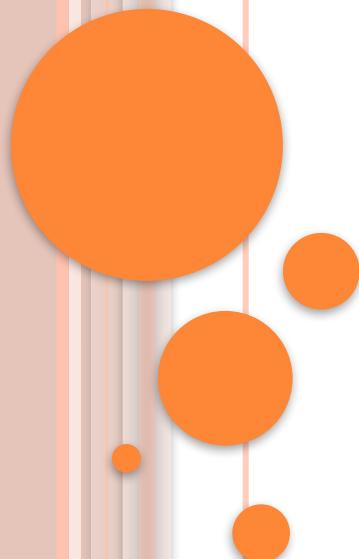
Tools ▾

Partner Content ▾



[RoR Tutorial.docx](#)

Rubgyn Rails



RAILS

extremely productive MVC framework for **database-backed** Web apps

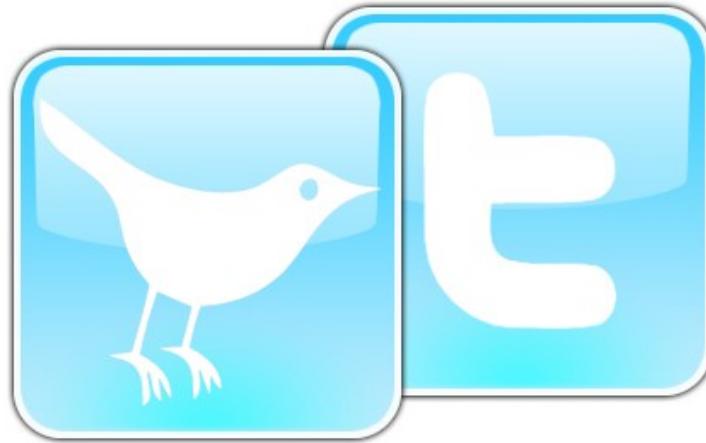
DRY - Don't Repeat Yourself!

CoC - Convention over Configuration

Manage **database** through **SQLite** and **migrations**



WHO USES **RUBY ON RAILS?**



Please clock in

First name:

Last name:



[Back](#)

See [payroll](#)

```
def login
  if request.get?
    #do nothing
  elsif request.post?
    @employee=Employee.new(params[:employee])
    @employee.time_in=Time.now
    entry= Employee.find(:first, :conditions => ["first_name = ? and last_name = ?",
                                                    @employee.first_name, @employee.last_name])
    if !entry.nil?
      if !entry.time_out.nil?
        entry.time_out=nil
        entry.pay=nil
      end
      entry.update_attributes(:time_in => @employee.time_in)
    elsif @employee.save
      #new entry created
    end
    redirect_to :action => "show"
  end
end
```



Users

Employee ID	First Name	Last Name	Time In	Time Out	Pay
-------------	------------	-----------	---------	----------	-----

1	Toan	Tran	03:50:20 AM	03:50:26 AM	\$45.00
3	Keith	Tran	03:00:00 AM	05:21:21 PM	\$9,607.00
7	Grant	Whittle	03:44:38 AM		

[Delete](#)[Delete](#)[Clock Out](#) [Delete](#)

Back to [clock in](#) page

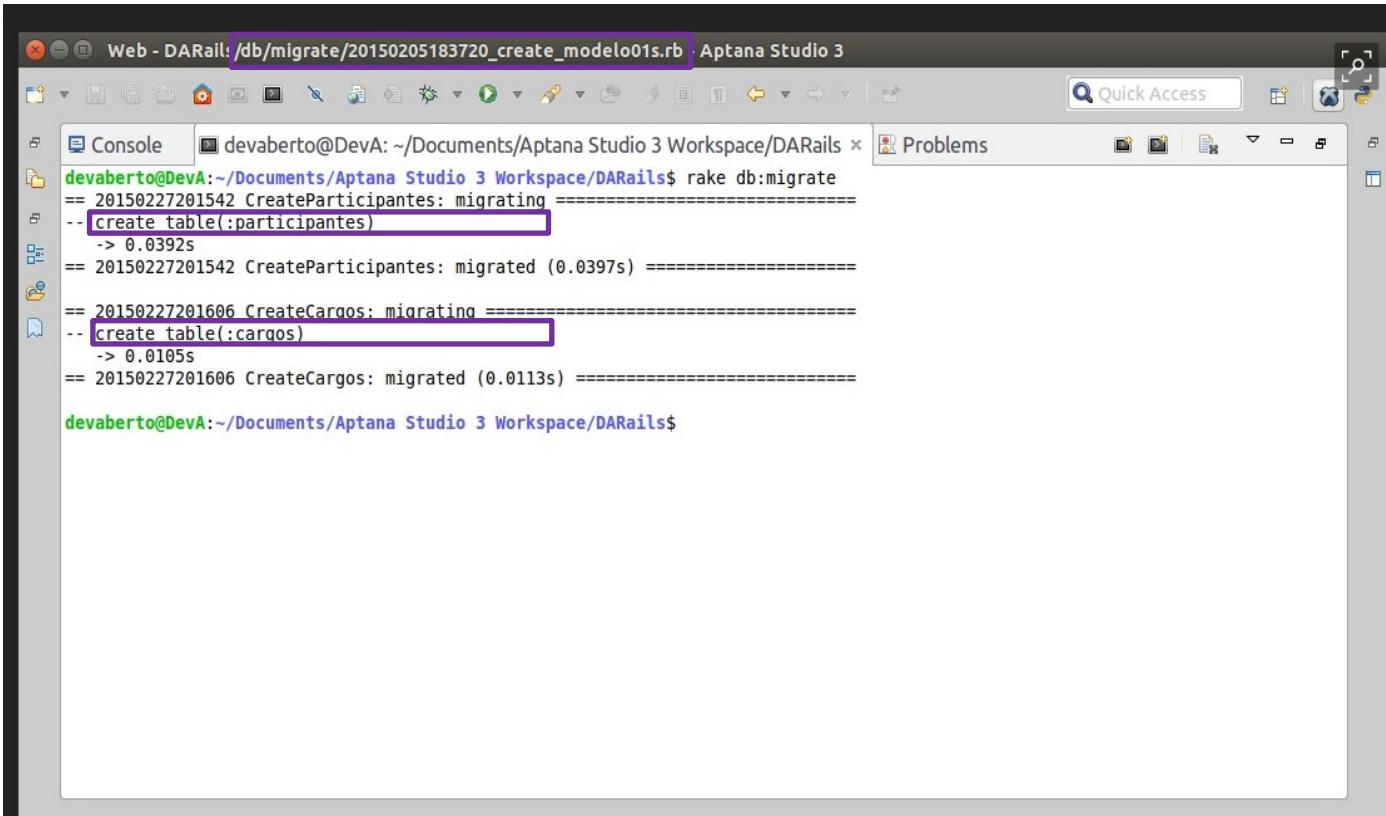
[Home](#)

```
<% @employees.each do |employee| %>
<tr>
  <td> <%= employee.id %> </td>
  <td> <%= employee.first_name %> </td>
  <td> <%= employee.last_name %> </td>
  <td> <%= employee.time_in.strftime("%I:%M:%S %p") %> </td>
  <td>
    <% if !employee.time_out.nil? %>
      <%= employee.time_out.strftime("%I:%M:%S %p") %>
    <% end %>
  </td>
  <td> <%= number_to_currency(employee.pay) %> </td>
  <td>
    <% if employee.time_out.nil?%>
      <%= link_to 'Clock Out', :action => 'clockout', :id => employee.id %>
    <% end %>
  </td>
  <td>
    <%= link_to 'Delete', :action => 'delete', :id => employee.id %>
  </td>
</tr>
<% end %>
```



```
1 class CreatePosts < ActiveRecord::Migration
2   def self.up
3     create_table :posts do |t|
4       t.string :title
5       t.text :body
6
7       t.timestamps
8     end
9   end
10
11  def self.down
12    drop_table :posts
13  end
14 end
15
```

M



The screenshot shows the Aptana Studio 3 interface with the title bar "Web - DARails /db/migrate/20150205183720_create_modelo01s.rb · Aptana Studio 3". The main area is the "Console" tab, which displays the following command-line output:

```
devaberto@DevA:~/Documents/Aptana Studio 3 Workspace/DARails$ rake db:migrate
== 20150227201542 CreateParticipantes: migrating =====
-- create_table(:participantes)
  -> 0.0392s
== 20150227201542 CreateParticipantes: migrated (0.0397s) =====

== 20150227201606 CreateCargos: migrating =====
-- create_table(:cargos)
  -> 0.0105s
== 20150227201606 CreateCargos: migrated (0.0113s) =====

devaberto@DevA:~/Documents/Aptana Studio 3 Workspace/DARails$
```

Using an MVC framework makes building web applications easy

Web Database Application

Development

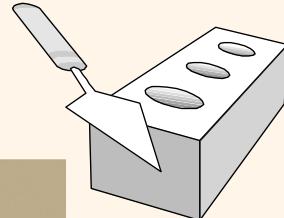
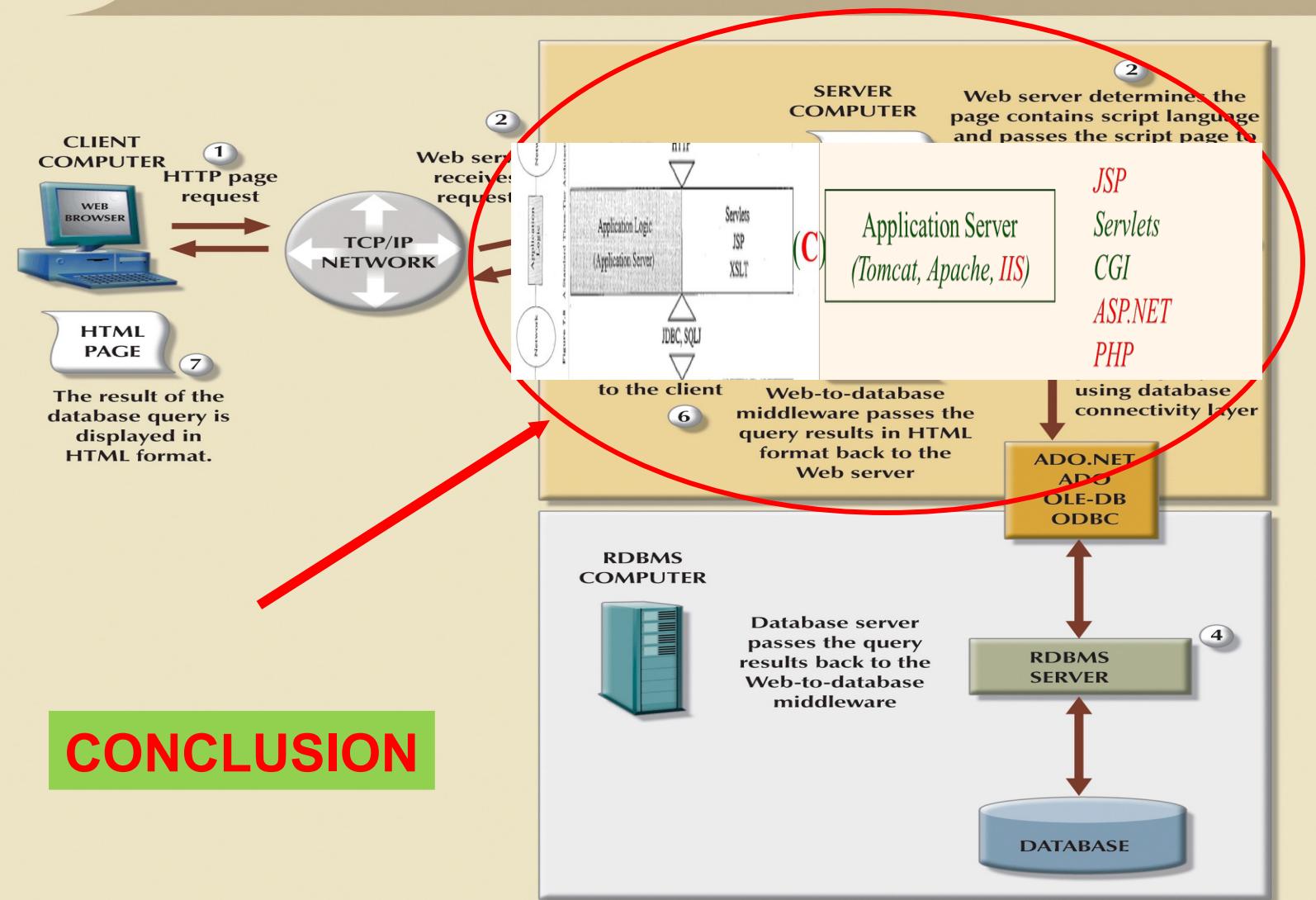


FIGURE
14.7

Web-to-database middleware



The only difference is the Application Server: C#, JAVA, PHP, RUBY, PYTHON

TUTORIAL 3 on AJAX, C# Visual Studio IDE MVC framework, PYTHON PyCharm IDE Django framework, PHP PhpStorm IDE Zend framework WEBB APPS

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾

HAD ENOUGH?

09.27.2023 (W 4 to 5:30) (11)	<p>Estimating and Planning:</p> <ul style="list-style-type: none">1. Estimating Techniques 2015.pdf2. How do you estimate on an Agile project 2015.pdf	<p>Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks WE CANVAS APPS</p>	<p>WEB APP Papers Summary (1 Page) CANVAS Assignment</p>	
---	--	---	---	--

10.02.2023 (M 4 to 5:30) (12)	<p>Lecture 4: Estimating and Planning</p> <p>Tutorials 4 COCOMO and MS Project</p>	<p>Estimating and Planning Papers Summary (1 Page) CANVAS Assignment</p>
---	--	---

09.27.2023
(W 4 to 5:30)

(11)

Estimating and Planning:
1. Estimating Techniques 2015.pdf
2. How do you estimate on an Agile project 2015.pdf

Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks WECANVAS APPS

WEB APP Papers
Summary
(1 Page)
CANVAS
Assignment

CLASSES

PAPERS on ESTIMATING and PLANNING	VH– Publish.
Estimating Techniques 2015.pdf	VH– Publish.
How do you estimate on an Agile project 2015.pdf	VH– Publish.

One Page Summary of Estimating and Planning Papers [A+](#)

- 12 pt. font, double spaced, MLA format, no citations required, times new roman.
- Introduction, body, and conclusion paragraphs.
- Submit as PDF or DOCX file.

Please attach your electronic version.

VH– Publish.

From 5:05 to 5:15 – 10 minutes.

09.27.2023 (W 4 to 5:30) (11)	Estimating and Planning: 1. Estimating Techniques 2015.pdf 2. How do you estimate on an Agile project 2015.pdf	Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks WECANVAS APPS	WEB APP Papers Summary (1 Page) CANVAS Assignment	
---	---	---	--	--

CLASS PARTICIPATION 20 points

20% of Total + :

PASSWORD: IN TEAMS

END Class 11 Participation

CLASS PARTICIPATION 20% Module | Not available until Sep 27 at 5:05pm | Due Sep 27 at 5:15pm | 100 pts

At 5:15.

End Class 11

VH, Download Attendance Report
Rename it:
9.27.2023 Attendance Report FINAL

VH, upload Class 11 to CANVAS.