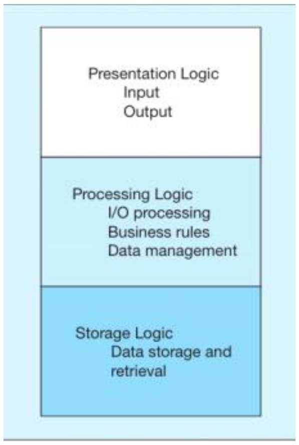
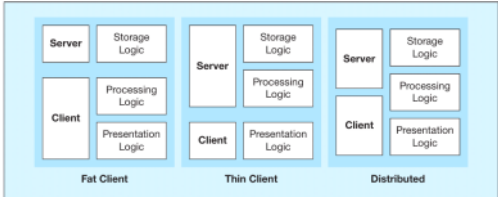
CLIENT/SERVER ARCHITECTURES

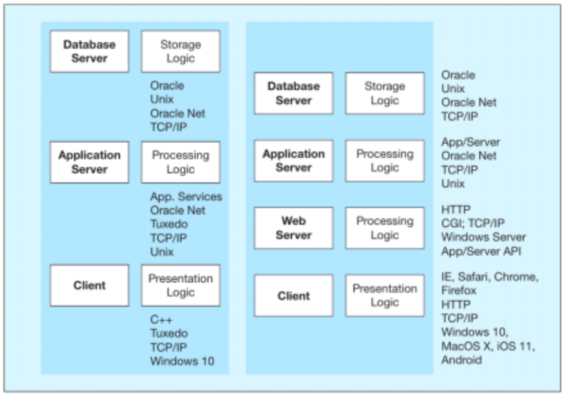
Client/server architectures can be distinguished by how application logic components are distributed across clients and servers.

- There are three components of application logic:
  - presentation logic: This component is responsible for formatting and presenting data on the user's screen or other output device and for managing user input from a keyboard or other input device (such as your phone or tablet's screen).
    - Presentation logic (think Web browser) resides on the client and is the mechanism with which the user interacts with the system.
  - processing logic: This handles data processing logic, business rules logic, and data management logic.
    - Data processing logic includes such activities as data validation and identification of processing errors.
    - Business rules that have not been implemented at the database management system (DBMS) level may be coded in the processing component.
    - Processing logic resides on both the client and servers.
  - storage logic: the component responsible for data storage and retrieval from the physical storage devices associated with the application.
    - Storage logic usually resides on the database server, close to the physical location of the data.
    - Activities of a DBMS occur in the storage logic component.
      - □ DFor example, data integrity control activities, such as constraint checking, are typically placed there.
      - □ Triggers, which will always fire when appropriate conditions are met, are associated with insert, modify, update, and delete commands, and they are also placed on the database server, as are stored procedures.



- Client/server architectures are normally categorized into three types: two-, three-, or n-tier architectures, depending on the placement of the three types of application logic (the three logics above).
  - No single optimal client/server architecture is the best solution for all business problems.
  - Figure 7-2a depicts three commonly found configurations of two-tier systems based on the placement of the processing logic.
    - 

Two tier system environment options.
In the fat client, the application processing occurs entirely on the client, whereas in the thin client, this processing occurs primarily on the server. In the distributed example, application processing is partitioned between the client and the server.

    - 

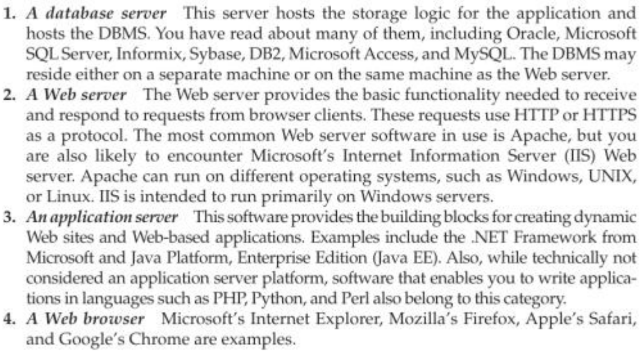
n-tier client server environments

These types of architectures are most prevalent in Web-based systems.

a typical client in a Web enabled client/server environment will be a thin client, using a browser or a smart phone app for its presentation logic.

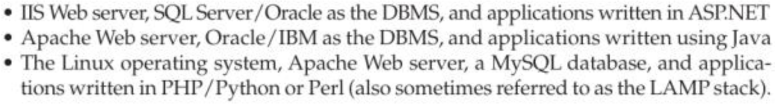
The middle tiers are typically coded in a portable language, such as C#, Java, or Python/PHP.

- Figure 7-2 shows the various components of a typical Web application. Four key components must be used together to create a Web application site:



1. *A database server* This server hosts the storage logic for the application and hosts the DBMS. You have read about many of them, including Oracle, Microsoft SQL Server, Informix, Sybase, DB2, Microsoft Access, and MySQL. The DBMS may reside either on a separate machine or on the same machine as the Web server.
2. *A Web server* The Web server provides the basic functionality needed to receive and respond to requests from browser clients. These requests use HTTP or HTTPS as a protocol. The most common Web server software in use is Apache, but you are also likely to encounter Microsoft's Internet Information Server (IIS) Web server. Apache can run on different operating systems, such as Windows, UNIX, or Linux. IIS is intended to run primarily on Windows servers.
3. *An application server* This software provides the building blocks for creating dynamic Web sites and Web-based applications. Examples include the .NET Framework from Microsoft and Java Platform, Enterprise Edition (Java EE). Also, while technically not considered an application server platform, software that enables you to write applications in languages such as PHP, Python, and Perl also belong to this category.
4. *A Web browser* Microsoft's Internet Explorer, Mozilla's Firefox, Apple's Safari, and Google's Chrome are examples.

- As you can see, a bewildering collection of tools are available to use for Web application development. However, the following are the most common combinations you will encounter:



- IIS Web server, SQL Server/Oracle as the DBMS, and applications written in ASP.NET
- Apache Web server, Oracle/IBM as the DBMS, and applications written using Java
- ▪ The Linux operating system, Apache Web server, a MySQL database, and applications written in PHP/Python or Perl (also sometimes referred to as the LAMP stack).

-

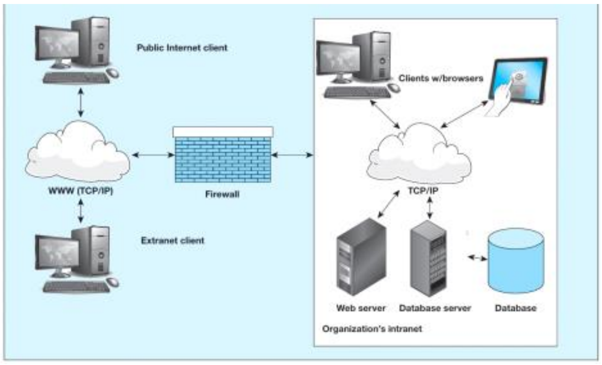
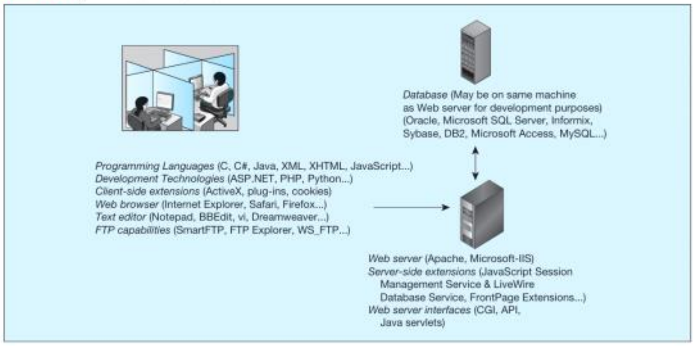


FIGURE 7-4 Dynamic Web development environment


Database (May be on same machine as Web server for development purposes) (Oracle, Microsoft SQL Server, Informix, Sybase, DB2, Microsoft Access, MySQL...)
Programming Languages (C, C#, Java, XML, XHTML, JavaScript...)
Development Technologies (ASP.NET, PHP, Python...)
Client-side extensions (ActiveX, plug-ins, cookies)
Web browser (Internet Explorer, Safari, Firefox...)
Text editor (Notepad, BBEdit, vi, Dreamweaver...)
FTP capabilities (SmartFTP, FTP Explorer, WS_FTP...)
Web server (Apache, Microsoft-IIS)
Server-side extensions (JavaScript Session Management Service & LiveWire Database Service, FrontPage Extensions...)
Web server interfaces (CGI, API, Java servlets)