THE UNIVERSITY OF MELBOURNE

Applications. Web Applications and Databases

Database Systems & Information Modelling
INFO90002

Week 7 – Web Applications
Dr Tanya Linden
Dr Greg Wadley
David Eccles

Today’s Session…

How end-users access the database

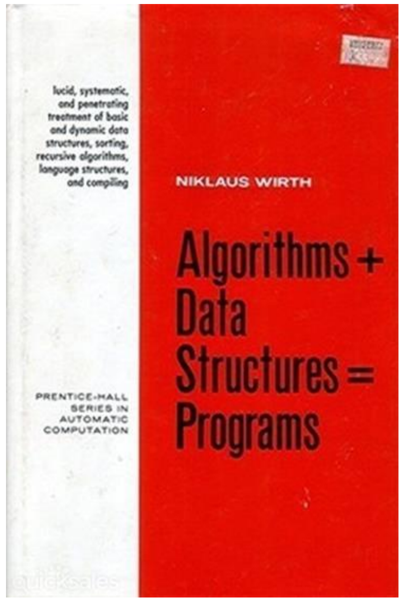
Business logic

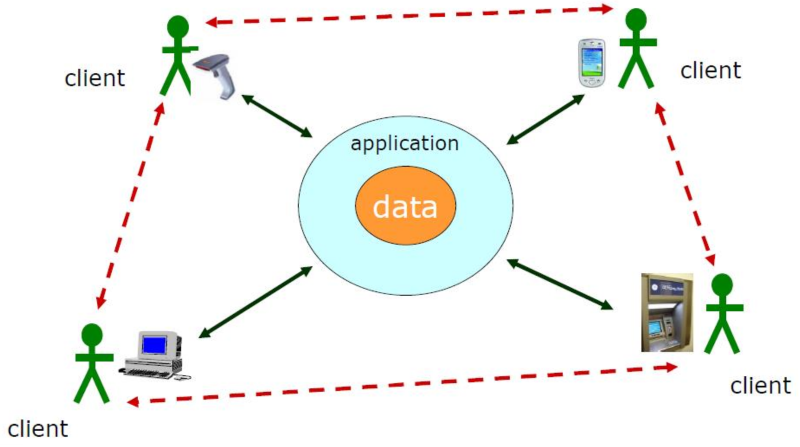
Embedding databases inside applications

Application architectures

Web applications

- How web apps work
- Making an HTML document
- Connecting to the DB
- Web services

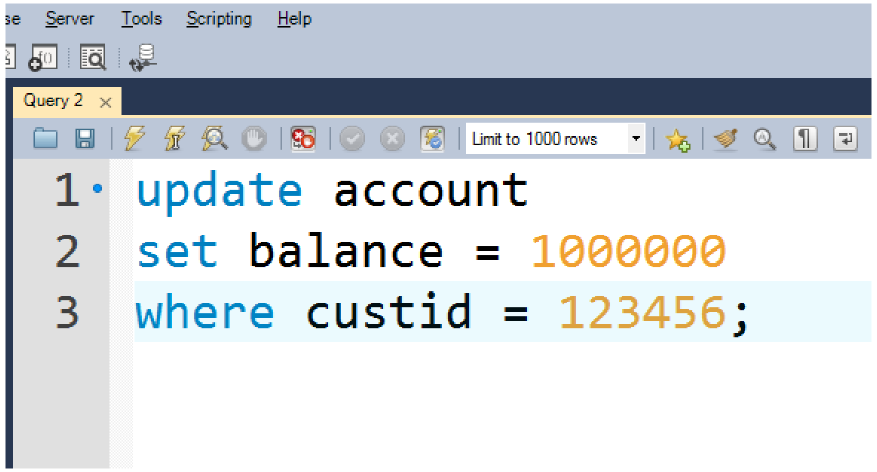

lucid, systematic, and penetrating treatment of basic and dynamic data structures, sorting, recursive algorithms, language structures, and compiling
NIKLAUS WIRTH
Algorithms + Data Structures = Programs
PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION



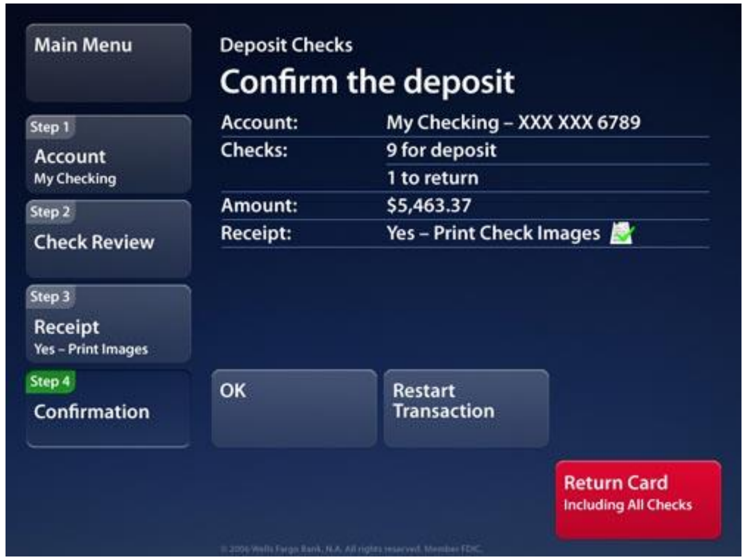
Limitations of SQL

SQL is declarative, intuitive, versatile, but …

- cannot express all possible queries in SQL
- need to enforce business rules beyond domain/referential integrity
- need procedural constructs such as loops and decisions
- would you give end-users a query browser? Why not?
- need a user interface that is both friendly and constraining



```
update account
set balance = 1000000
where custid = 123456;
```



Deposit Checks

Confirm the deposit

Account:	My Checking – XXX XXX 6789
Checks:	9 for deposit
	1 to return
Amount:	$5,463.37
Receipt:	Yes – Print Check Images

How to handle business logic?

Examples of business logic:

- Check name and password. If good, login, if bad, error message
- Insert one row in *Order* table, then several in *OrderItem* table
- Check amount < balance. If so, subtract amount from one row in bank account table, then add amount to another row
- For all rows in Customer table, send out monthly statements

Procedural programming languages can do:

- Sequence (several steps performed in order)
- Iteration (loops)
- Control flow (conditionals, decisions)
- User interface (accept input and present output for users)

SQL is specialized for low-level data access

Example business logic

Customer places an order

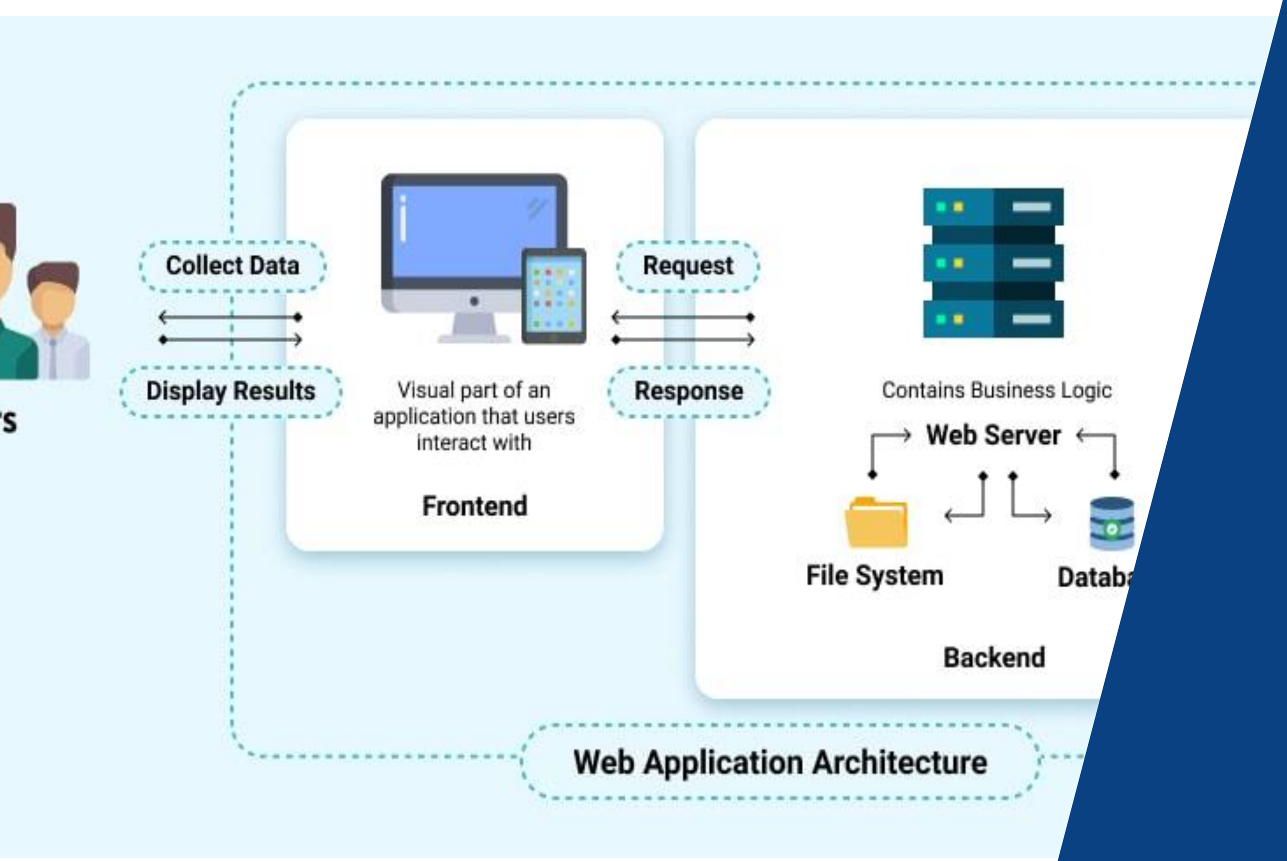
- Accept inputs from user (e.g. via web form)
- Insert row into Order table
- Repeat for each product ordered:
  - Check Product table shows sufficient quantity in stock. If so:

    Insert one row into OrderItem table

    Change Product table in-stock, Customer table amount-owing
- If no errors encountered, end successfully

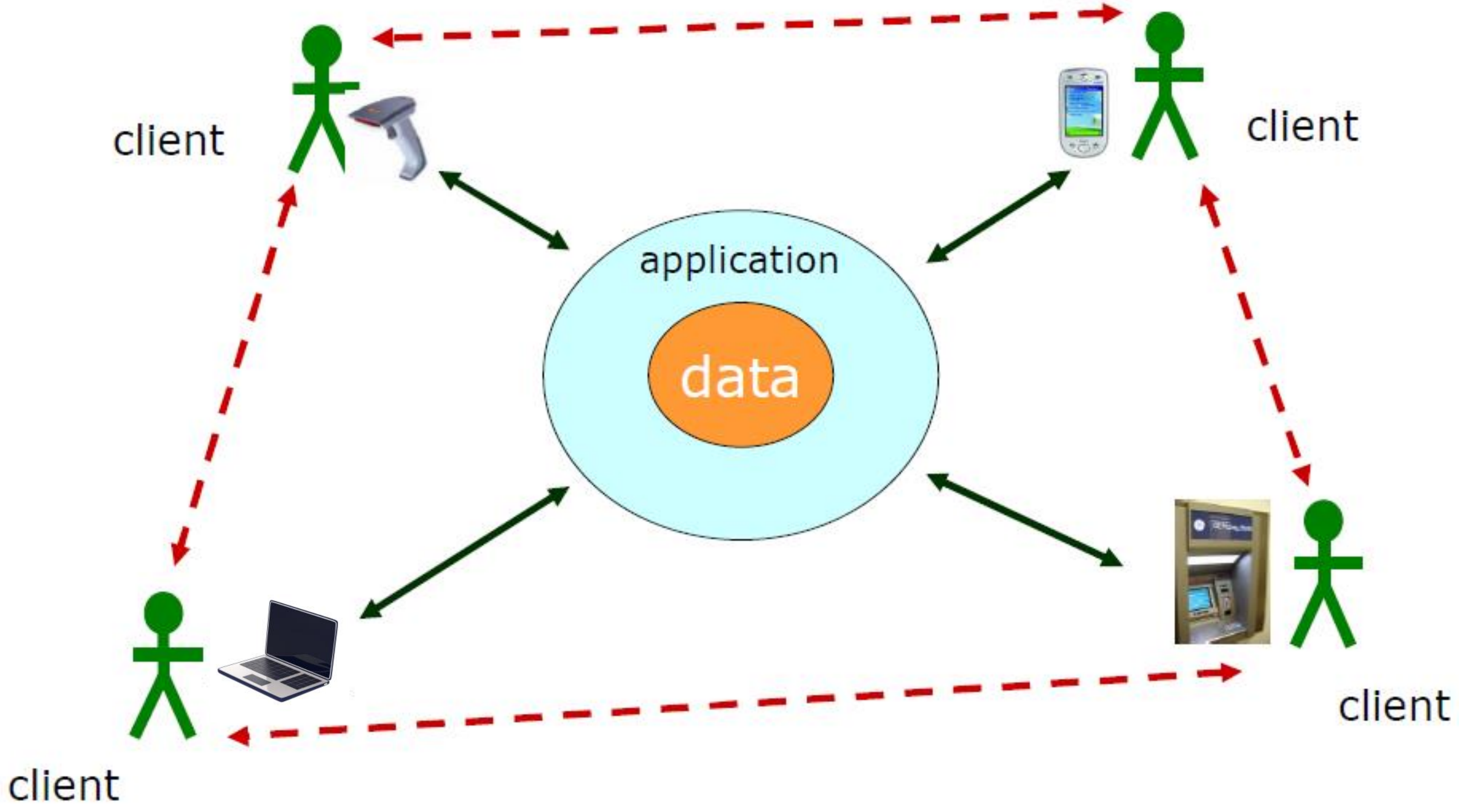
Customer moves money from savings to credit card account

- Accept inputs from user (via ATM, internet banking or mobile app)
- Select balance from savings account
- Is there enough money to withdraw? If so:
  - Update savings account balance = balance – withdrawal
  - Update credit card balance = balance + withdrawal
- If no errors encountered, end successfully



Application Architectures

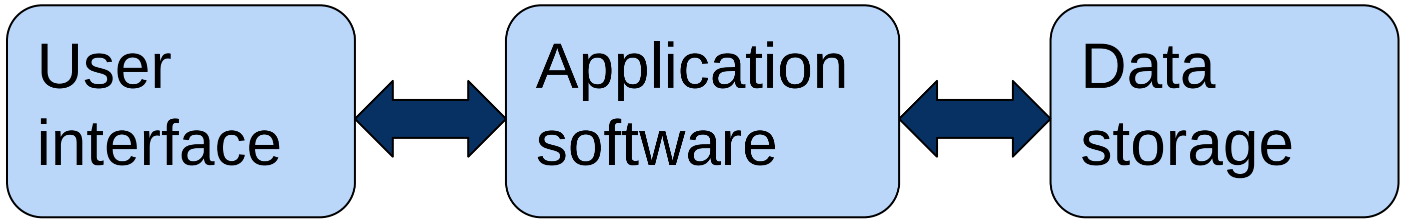
System architecture



System architecture

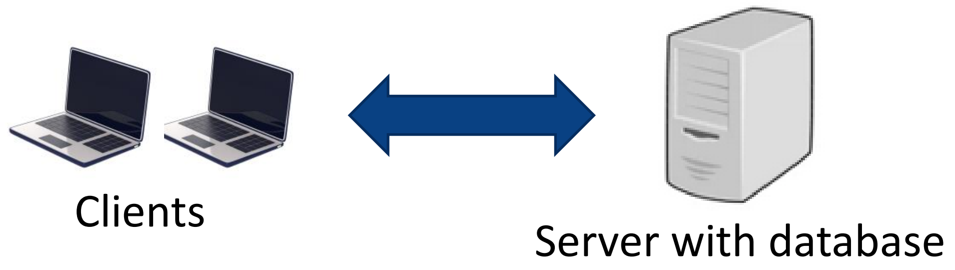
An information system must provide

- Presentation logic
  - input (keyboard, touchscreen, voice, sensor etc.)
  - output (large screen, printer, phone, ATM etc.)
- Business logic
  - input and command handling
  - enforcement of business rules
- Storage logic
  - persistent storage of data
  - enforcement of data integrity

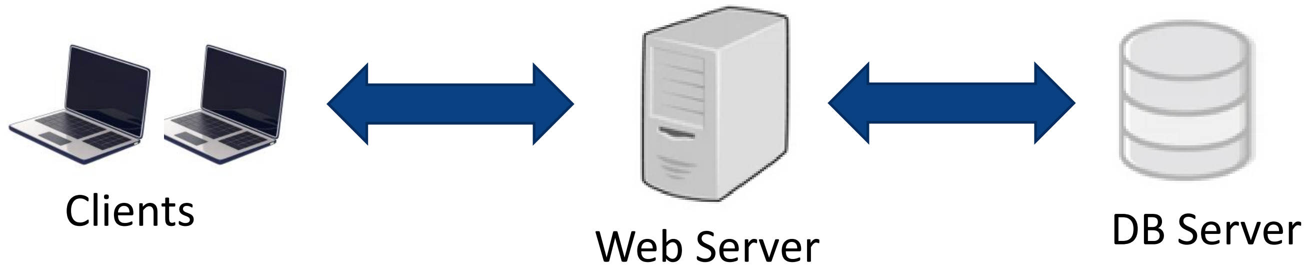


Multi-tiered architectures

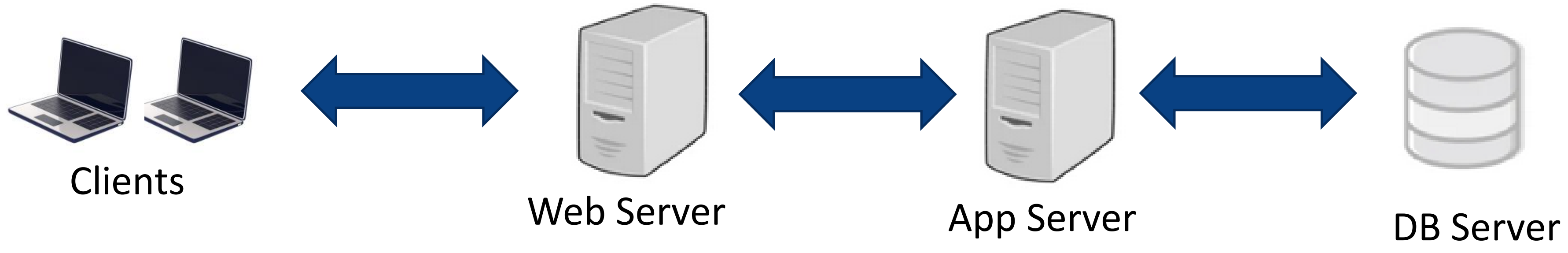
- 2 tiers



- 3 tiers



- 4 tiers



Evolution of application architectures

Mainframe / dumb terminal

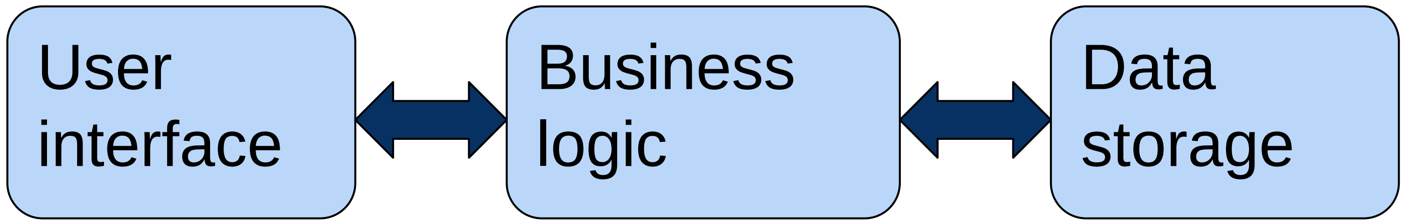
- One large computer handles all logic
- Problems: doesn't scale with number of users

Client-Server architecture

- 2-tier: e.g. file server, database, web
- 3-tier: separation of Presentation, Processing and Storage logic

Web architecture

- a particular form of 3 or 4 tier architecture



Mainframe ("1Tier")

Mainframes and mini-computers

Dumb terminals (no processing at client end)

Entire application ran on the same computer

- Database
- Business logic
- User interface

Enabling technologies included:

- Embedded SQL
- Report generators

Client Server - 2 Tier

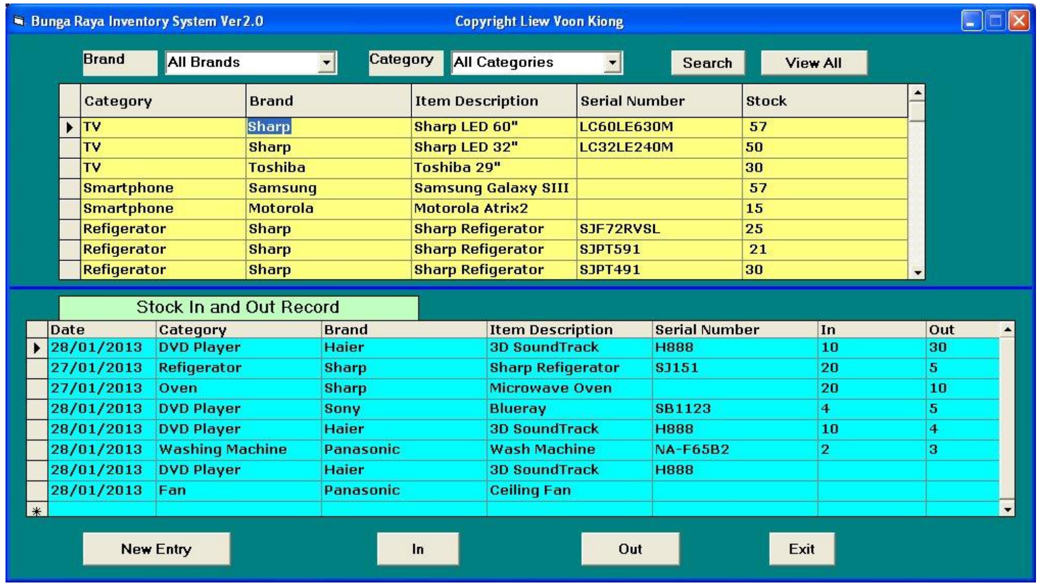
Server is a relational DBMS

- data storage and access is done at the DBMS

SQL queries sent to DB server, which returns raw data

Presentation, business logic is handled in client application

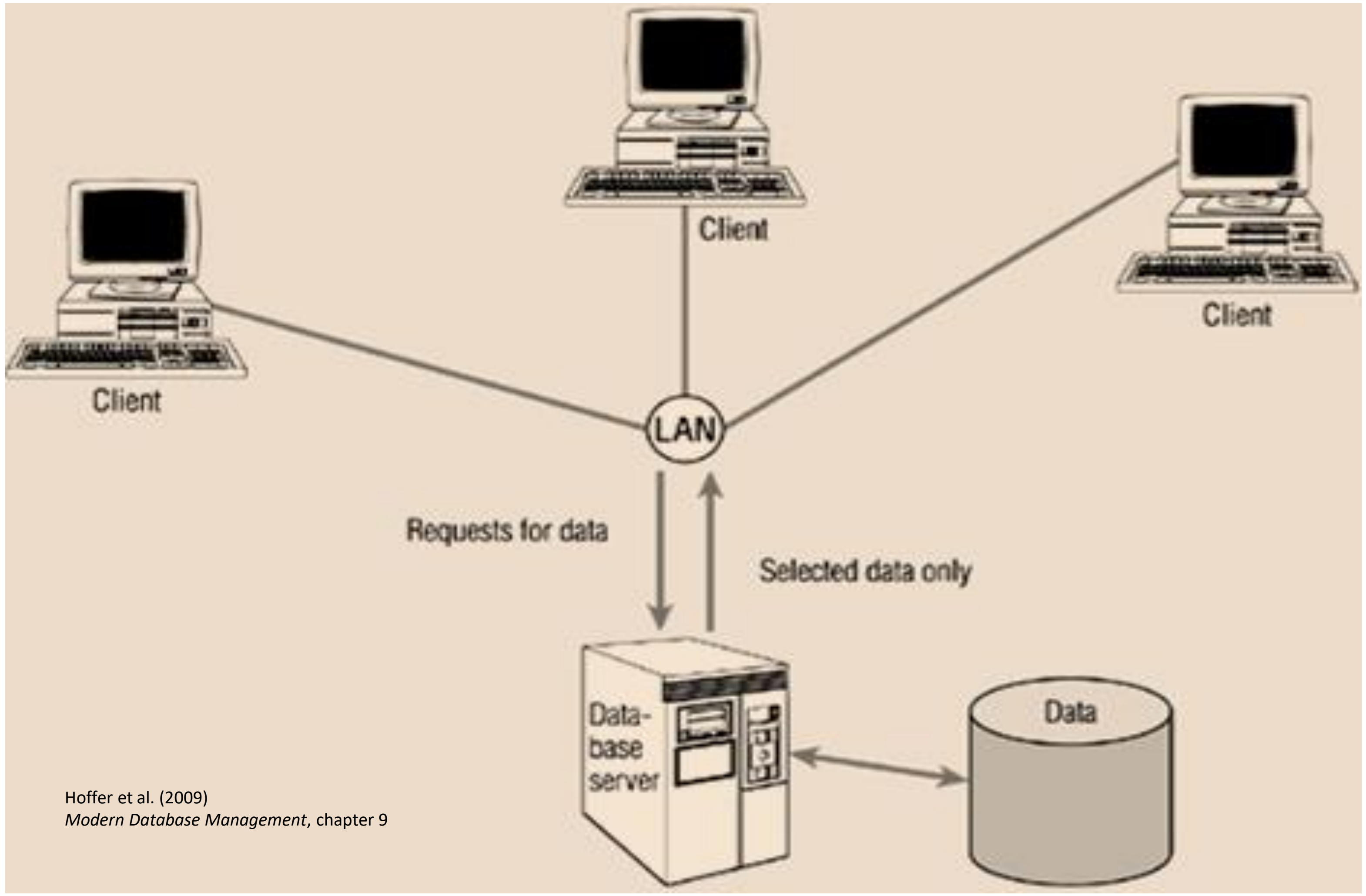
Platforms like Visual Basic (1990s into 2000s)



Category	Brand	Item Description	Serial Number	Stock
TV	Sharp	Sharp LED 60"	LC60LE630M	57
TV	Sharp	Sharp LED 32"	LC32LE240M	50
TV	Toshiba	Toshiba 29"		30
Smartphone	Samsung	Samsung Galaxy SIII		57
Smartphone	Motorola	Motorola Atrix2		15
Refigerator	Sharp	Sharp Refigerator	SJF72RVSL	25
Refigerator	Sharp	Sharp Refigerator	SJPT591	21
Refigerator	Sharp	Sharp Refigerator	SJPT491	30

Date	Category	Brand	Item Description	Serial Number	In	Out
28/01/2013	DVD Player	Haier	3D SoundTrack	H888	10	30
27/01/2013	Refigerator	Sharp	Sharp Refigerator	SJ151	20	5
27/01/2013	Oven	Sharp	Microwave Oven		20	10
28/01/2013	DVD Player	Sony	Blueray	SB1123	4	5
28/01/2013	DVD Player	Haier	3D SoundTrack	H888	10	4
28/01/2013	Washing Machine	Panasonic	Wash Machine	NA-F65B2	2	3
28/01/2013	DVD Player	Haier	3D SoundTrack	H888		
28/01/2013	Fan	Panasonic	Ceiling Fan			

2 Tier Example

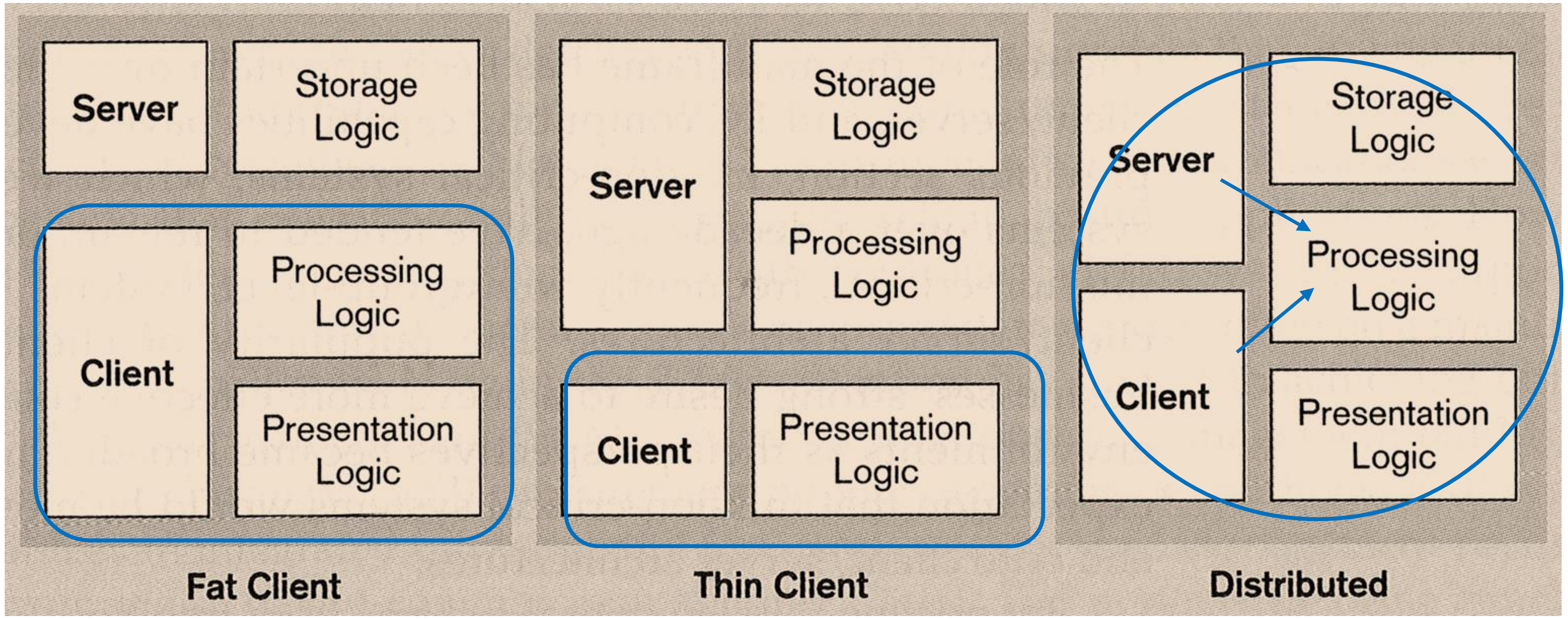


Hoffer et al. (2009)
Modern Database Management, chapter 9

Distribution of Processing Logic

2-tier distributions

- Processing logic could be at client, server, or both



Hoffer et al. (2009)
Modern Database Management, chapter 9

2-Tier advantages and disadvantages

Advantages

- Clients and server share processing load
- Good data integrity since data is all processed centrally
- Stored procedures allow some business rules to be implemented on the database server

Disadvantages

- Presentation, data model, business logic are intertwined at client
- If DB schema changes, all clients break
- Updates need to be deployed to all clients
- DB connection for every client, thus difficult to scale
- Difficult to implement beyond the organization (to customers)
- Interoperability issues

3-Tier architecture

Client program <-> Application server <-> Database server

Presentation logic

- Client handles interface
  - Thinner clients

    Limited or no data storage (possibly no hard disk)

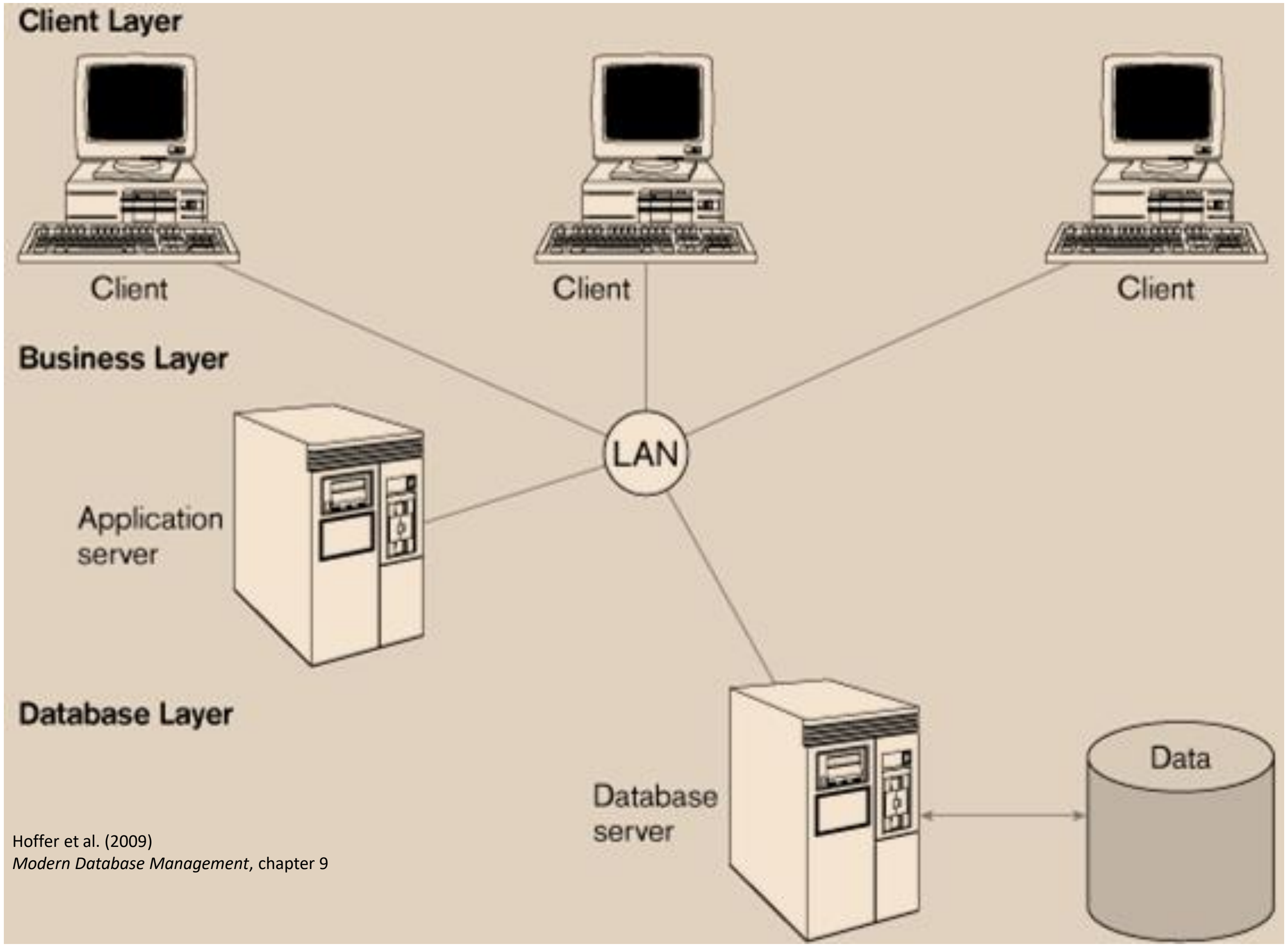
Business logic

- Application Server deals with business logic

Storage logic

- Database server deals with data persistence and access

A Three-tier architecture - Example



Hoffer et al. (2009)
Modern Database Management, chapter 9

3-Tier advantages and disadvantages

Advantages

- Scalability
- Technological flexibility (can change business logic easily)
- Can swap out any single component fairly easily
- Long-term cost reduction
- Improved security – customer machine does presentation only

Disadvantages

- High short-term costs
- Tools and training
- Complex to design
- Variable standards

3-Tier (web based –)

Browser handles presentation logic

Browser talks to web server via simple, standard protocol

Business logic and data storage handled on server(s)

Pros

- Everyone has a browser
- No need for install and maintain client software
- HTML and HTTP are simple standards, widely supported
- Opens up the possibility of global access to database

Cons

- Even more complexity in the middle-tier
- Simple standards = hard to make complex application
- Global access = potential security nightmare (next page)

Security in multi-tier applications

Network environment creates complex security issues

Security can be enforced at different tiers:

- application password security
  - – for allowing access to the application software
- database-level password security
  - – for determining access privileges to tables
- secure client/server communication
  - – via encryption

THE UNIVERSITY OF MELBOURNE

Web Applications

Overview of Web Apps

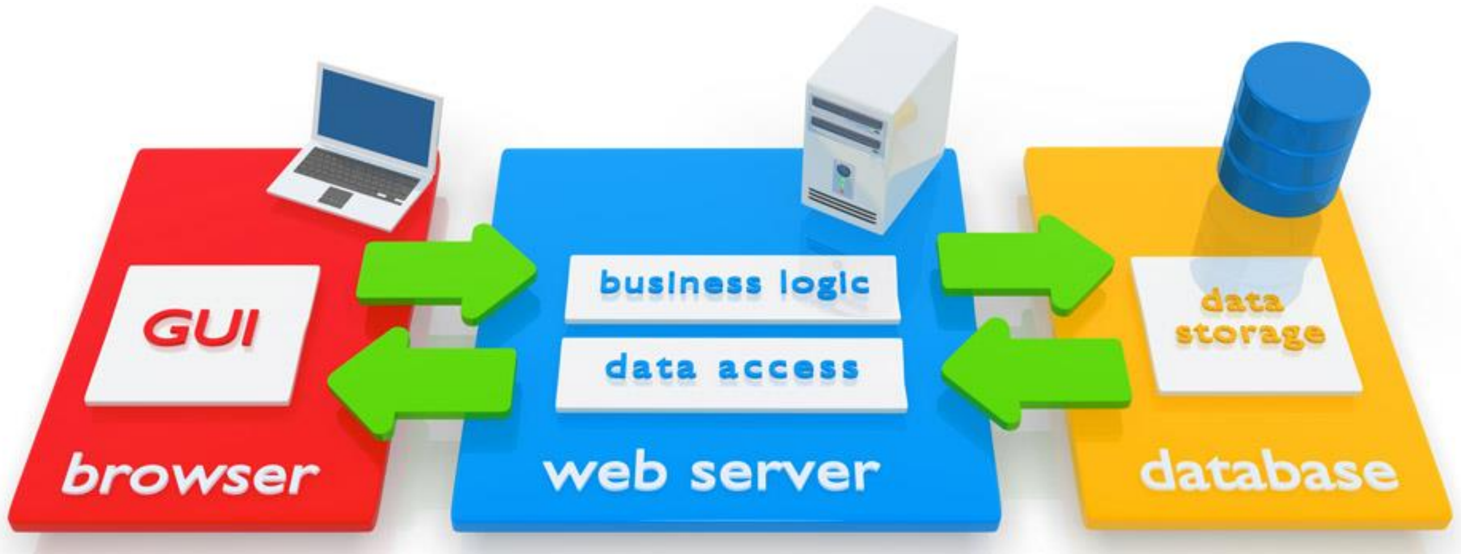
Why web apps?

How web apps work

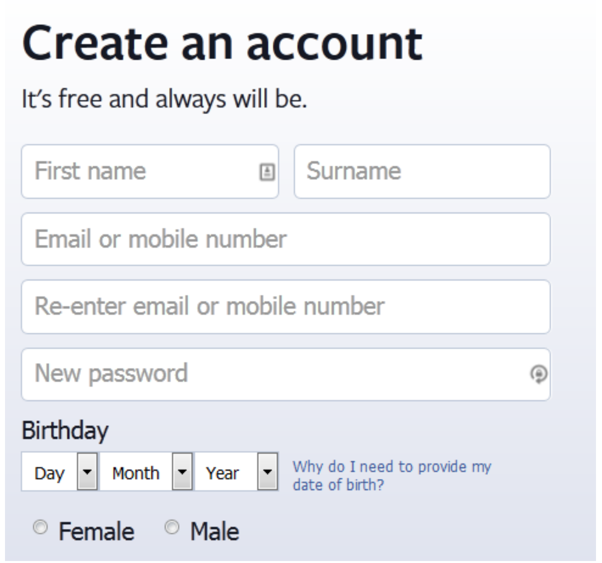
Making an HTML document

Connecting to the DB

Web services



Example web applications



Create an account

It's free and always will be.

First name | Surname

Email or mobile number

Re-enter email or mobile number

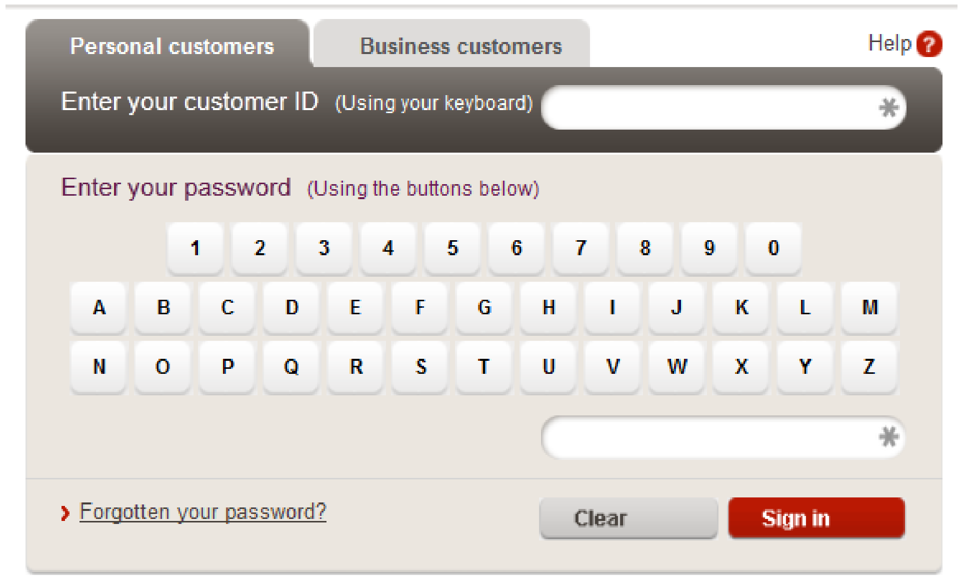
New password

Birthday

Day | Month | Year

Why do I need to provide my date of birth?

Female Male



Personal customers | Business customers | Help

Enter your customer ID (Using your keyboard)

Enter your password (Using the buttons below)

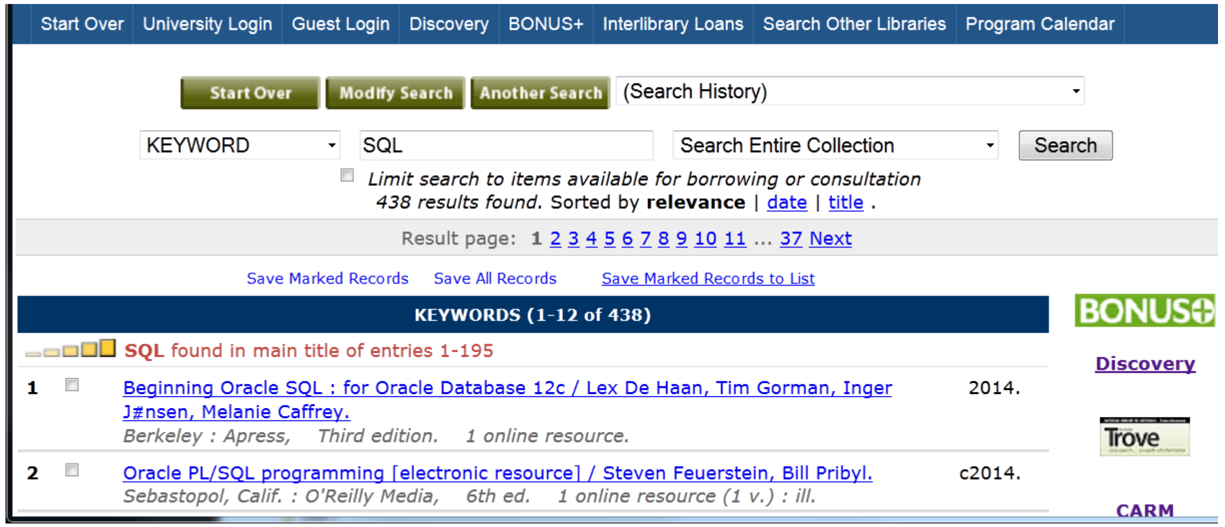
1 2 3 4 5 6 7 8 9 0

A B C D E F G H I J K L M

N O P Q R S T U V W X Y Z

Forgotten your password?

Clear | Sign in



Start Over | University Login | Guest Login | Discovery | BONUS+ | Interlibrary Loans | Search Other Libraries | Program Calendar

Start Over | Modify Search | Another Search | (Search History)

KEYWORD | SQL | Search Entire Collection | Search

Limit search to items available for borrowing or consultation

438 results found. Sorted by ***relevance*** | date | title .

Result page: 1 2 3 4 5 6 7 8 9 10 11 ... 37 Next

Save Marked Records Save All Records Save Marked Records to List

KEYWORDS (1-12 of 438)

SQL found in main title of entries 1-195

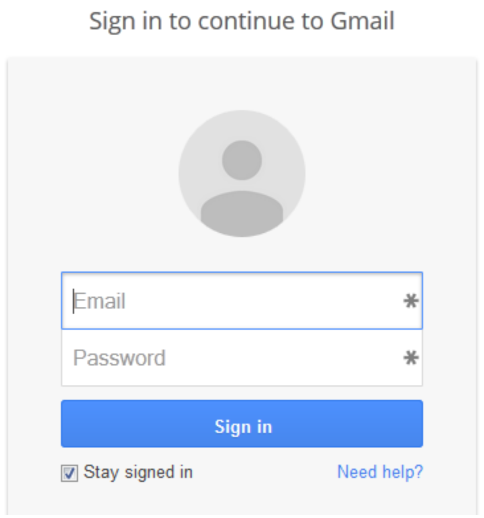
1		Beginning Oracle SQL : for Oracle Database 12c / Lex De Haan, Tim Gorman, Inger J#nsen, Melanie Caffrey. *Berkeley : Apress, Third edition. 1 online resource.*	2014.
2		Oracle PL/SQL programming [electronic resource] / Steven Feuerstein, Bill Pribyl. *Sebastopol, Calif. : O'Reilly Media, 6th ed. 1 online resource (1 v.) : ill.*	c2014.

BONUS+

Discovery

Trove

CARM



Sign in to continue to Gmail

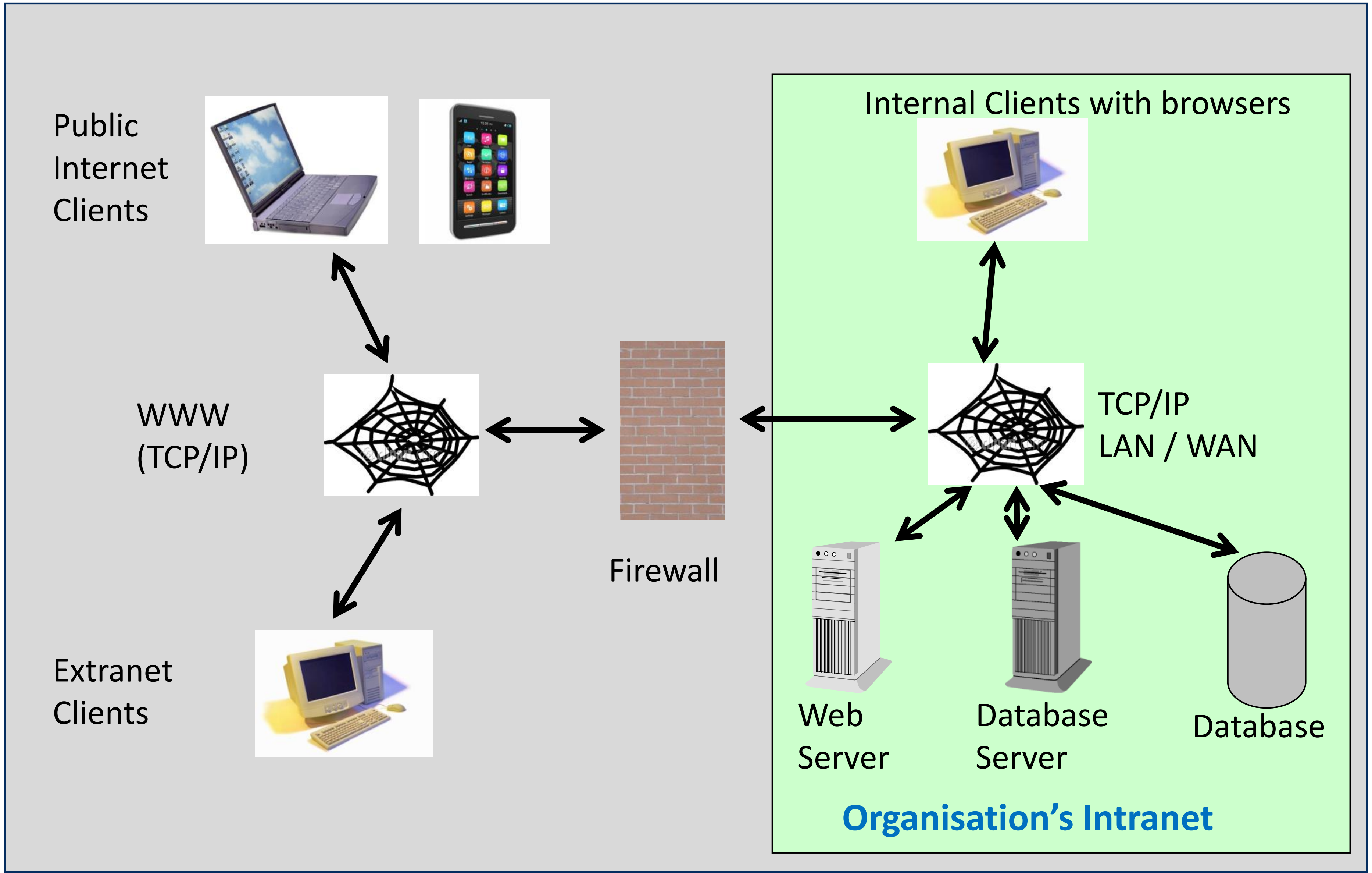
Email

Password

Sign in

Stay signed in Need help?

Architecture of a web app



Why create web applications?

Web browsers are ubiquitous

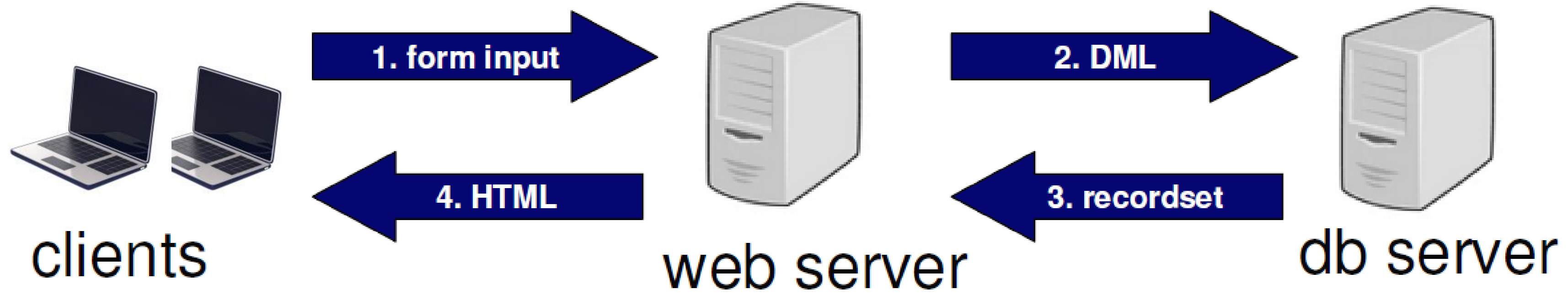
No need to install client software for external customers

Simple communication protocols

Platform and Operating System independent ("interoperable")

Reduction in development time and cost

Has enabled eGov, eBusiness, eCommerce, B2B, B2C



Web infrastructure

Browser

- Software that retrieves and displays HTML documents

Web Server

- Software that responds to requests from browsers by transmitting HTML and other documents to browsers

Web pages (HTML documents)

- Static web pages
  - content established at development time
- Dynamic web pages
  - content dynamically generated using data from database

World Wide Web (WWW)

- The total set of interlinked hypertext documents residing on Web servers worldwide

Internet

- Global network infrastructure that hosts the WWW

Web-related languages

Hypertext Markup Language (HTML)

- Markup language used to define a web page (content and structure)

Cascading Style Sheets (CSS)

- Control appearance of an HTML document

JavaScript (JS)

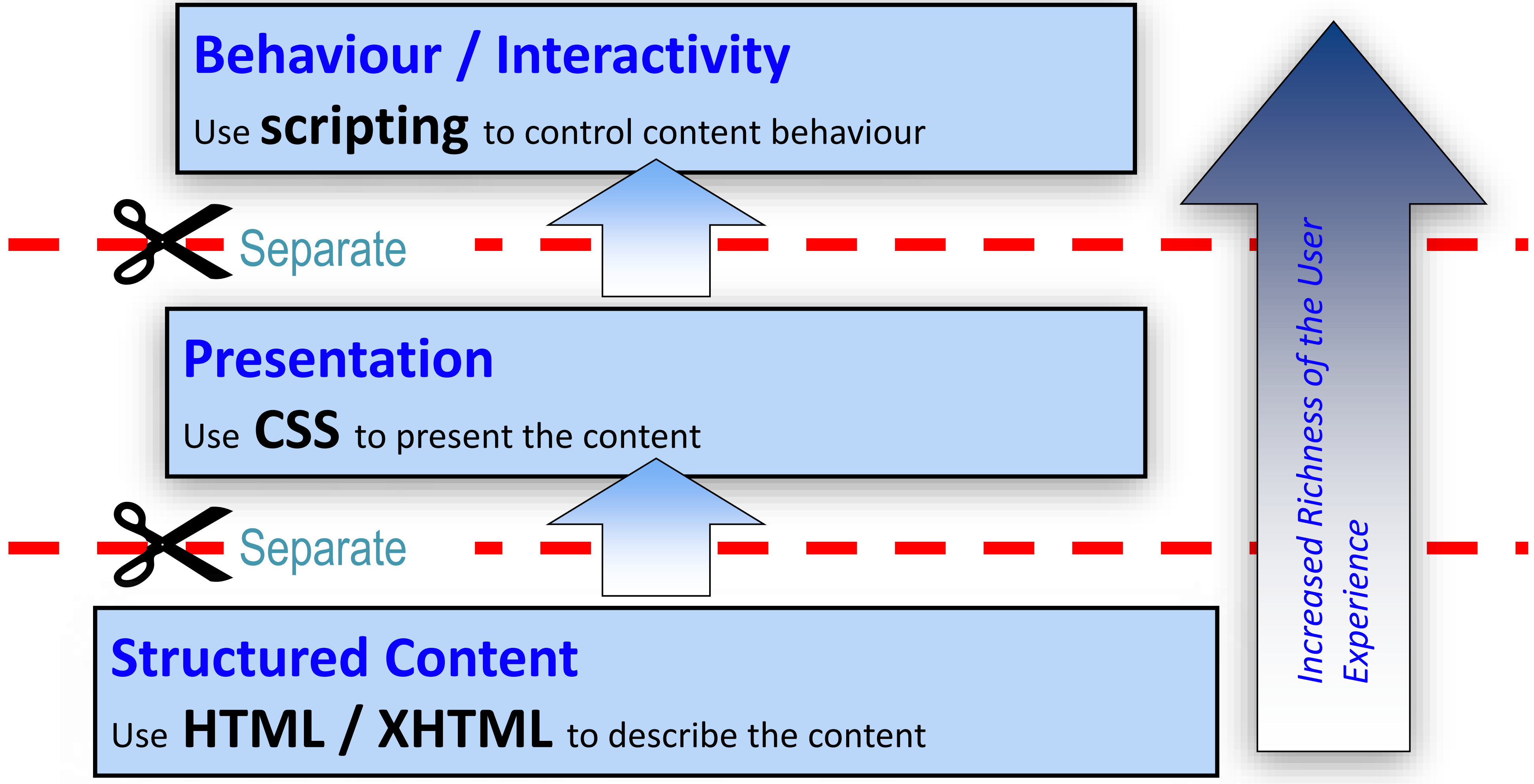
- Scripting language that enable interactivity in HTML documents

Extensible Markup Language (XML)

- Markup language used to transport data between web services

For more info www.w3schools.com

Build your webpages using the correct tools



Behaviour / Interactivity

Use **scripting** to control content behaviour

Presentation

Use **CSS** to present the content

Structured Content

Use **HTML / XHTML** to describe the content

Work from the bottom up !

Web page = HTML document

A structured file of elements defined by HTML tags

Interpreted by web browser for display

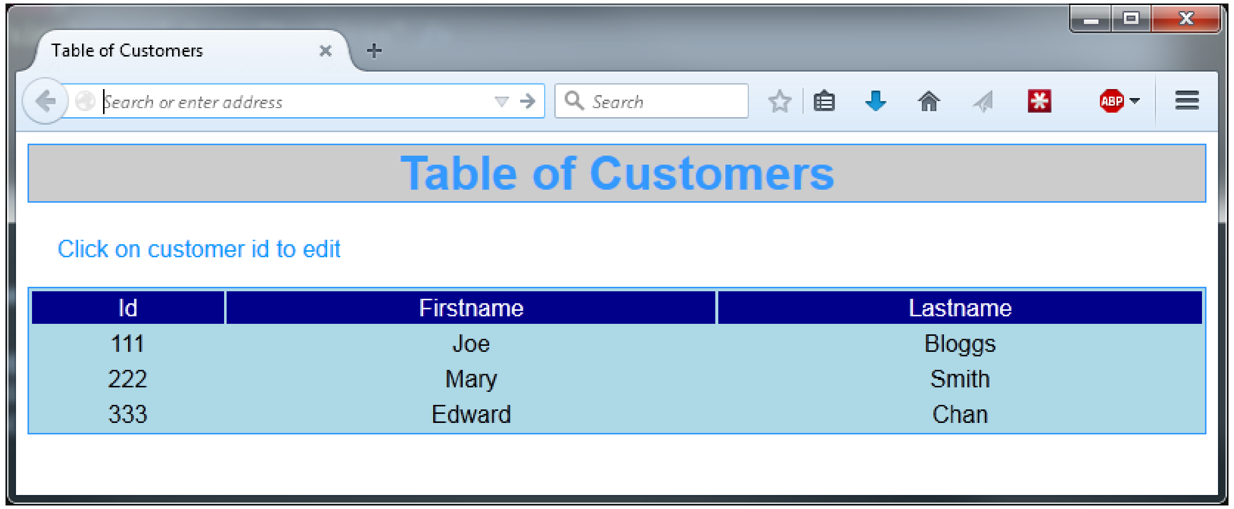
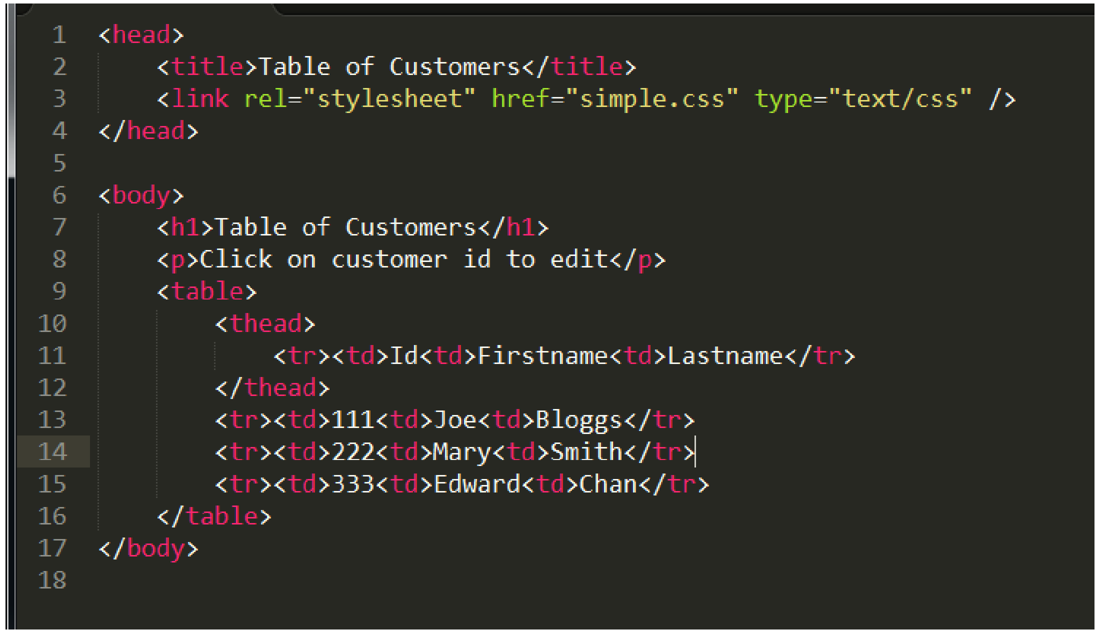


Table of Customers

Click on customer id to edit

Id	Firstname	Lastname
111	Joe	Bloggs
222	Mary	Smith
333	Edward	Chan

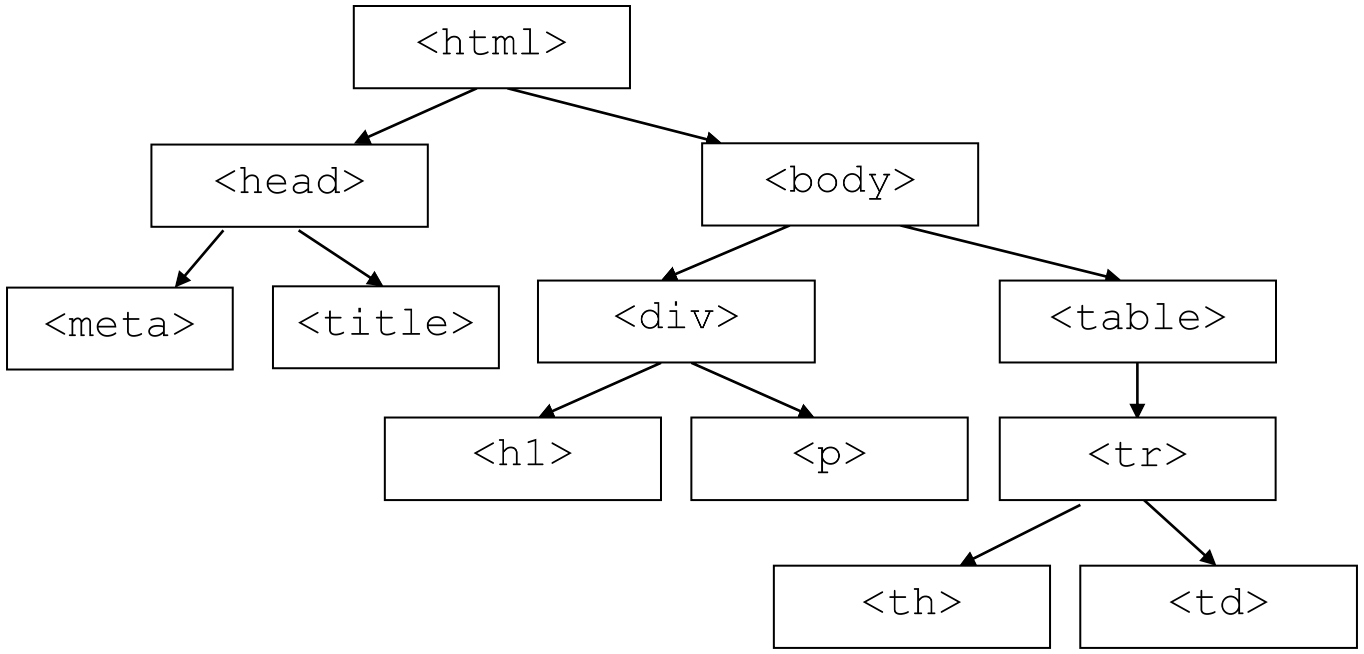

```
<head>
    <title>Table of Customers</title>
    <link rel="stylesheet" href="simple.css" type="text/css" />
</head>

<body>
    <h1>Table of Customers</h1>
    <p>Click on customer id to edit</p>
    <table>
        <thead>
            <tr><td>Id<td>Firstname<td>Lastname</tr>
        </thead>
        <tr><td>111<td>Joe<td>Bloggs</tr>
        <tr><td>222<td>Mary<td>Smith</tr>
        <tr><td>333<td>Edward<td>Chan</tr>
    </table>
</body>

```

HTML Document Structure – Tree View

The “**root**” element of any html document, is the **`html`** element, which usually contains only two children **`head`** and **`body`**

- The **`head`** then contains the **`title`**, and other ‘head’ elements.
- The **`body`** can contain many other elements

```
<html lang=“en”>
<head>
    <meta ....  />
    <title>...</title>
</head>
<body>
  <div>
    <h1>...</h1>
    <p>...</p>
  </div>
  <table>
    <tr>
         <th>...</th>
         <td>...</td>
    </tr>
  </table>
</body>
</html>
```



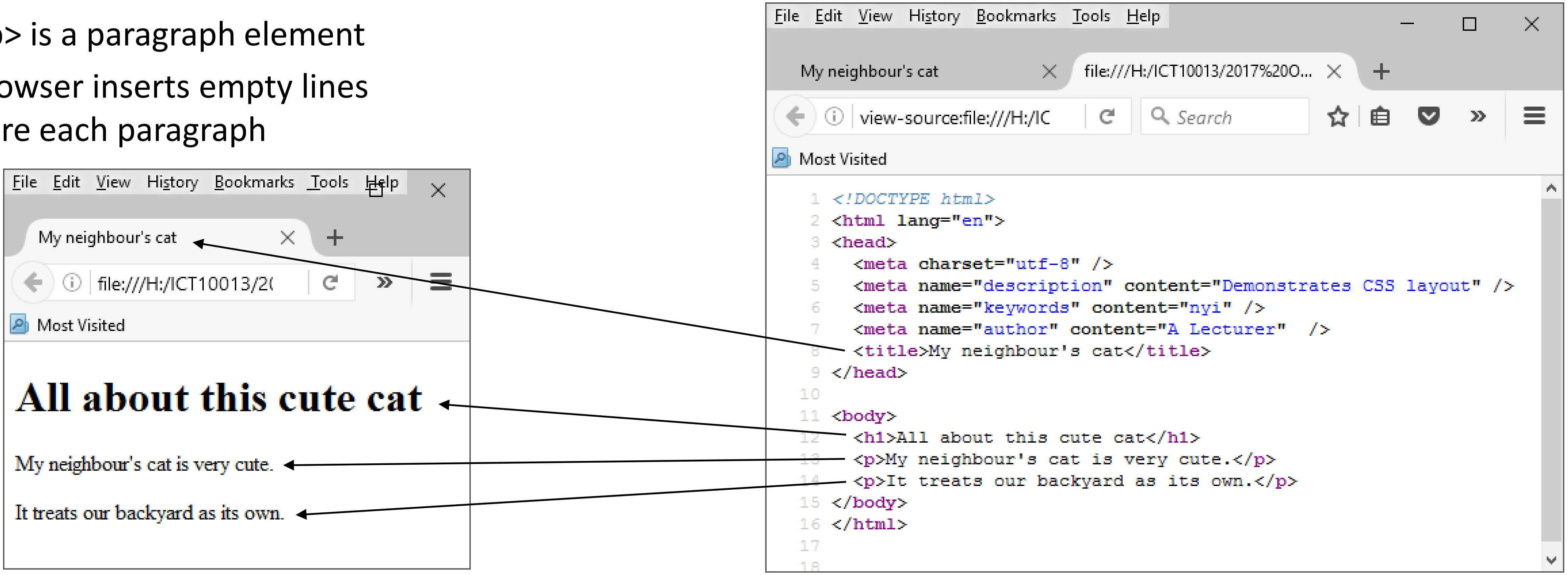
Web Browsers display basic web page

The <h1> is a heading element

- there are six heading sizes ranging from <h1> to <h6>
- <h6> is the smallest

The <p> is a paragraph element

- A browser inserts empty lines before each paragraph


File Edit View History Bookmarks Tools Help
My neighbour's cat
file:///H:/ICT10013/2(
Most Visited
file:///H:/ICT10013/2017%20O...
view-source:file:///H:/IC
Search

All about this cute cat

My neighbour's cat is very cute.

It treats our backyard as its own.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="description" content="Demonstrates CSS layout" />
  <meta name="keywords" content="nyi" />
  <meta name="author" content="A Lecturer"  />
  <title>My neighbour's cat</title>
</head>

<body>
  <h1>All about this cute cat</h1>
  <p>My neighbour's cat is very cute.</p>
  <p>It treats our backyard as its own.</p>
</body>
</html>
```

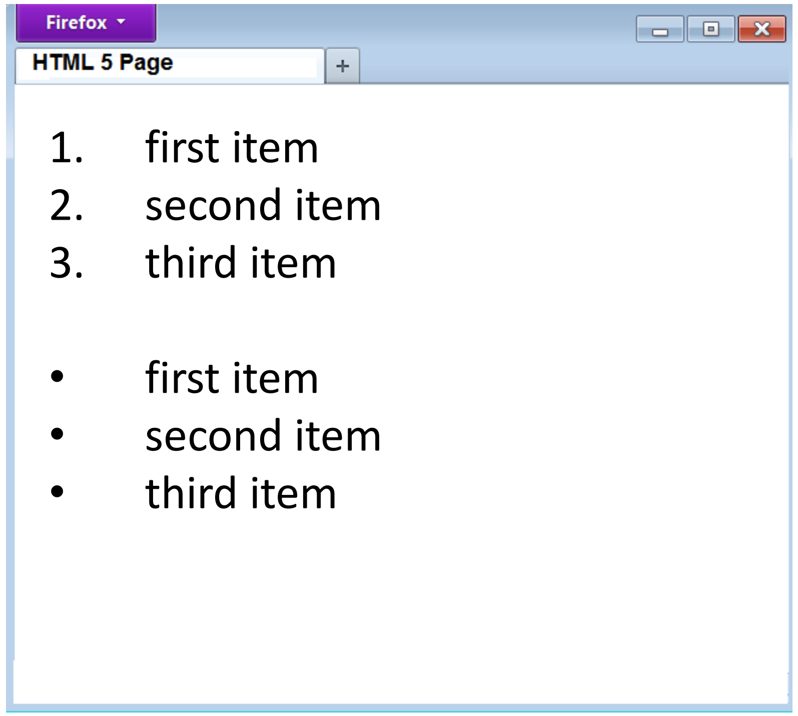
Lists

Ordered list example

```
<ol>
   <li>first item</li>
   <li>second item</li>
   <li>third item</li>
</ol>
```

Unordered list example

```
<ul>
   <li>first item</li>
   <li>second item</li>
   <li>third item</li>
</ul>
```



1. first item
2. second item
3. third item

- first item
- second item
- third item

Table

Deprecated attribute border = can now only be "1" (show a border) or "0" (do not show a border). Can style better in CSS.

```
<table>
  <caption>Table of Monthly Savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
  <tr>
    <td>Total</td>
    <td>$180</td>
  </tr>
</table>
```

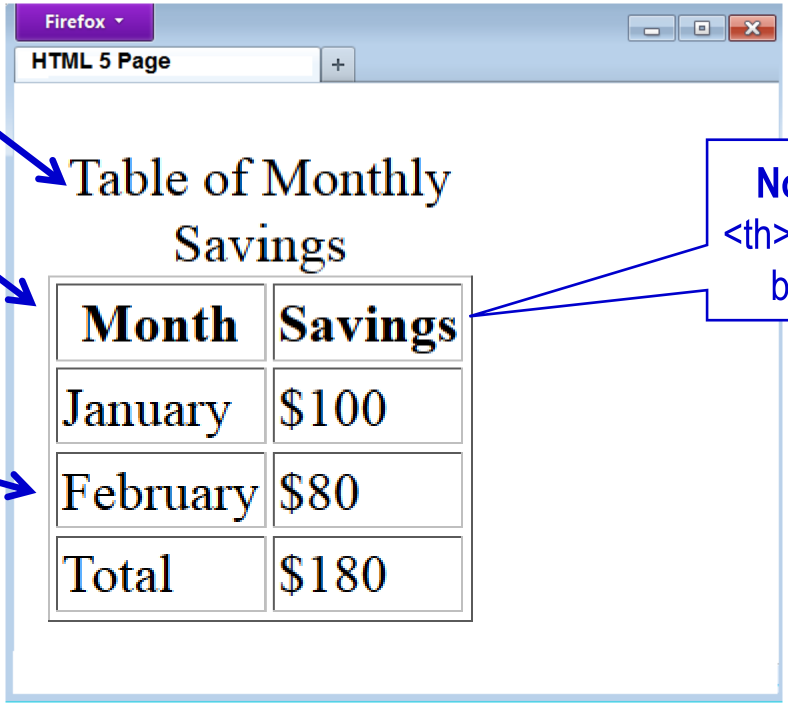


Table of Monthly Savings

Month	**Savings**
January	$100
February	$80
Total	$180

Note: by default the <th> cells are presented bold and centred !

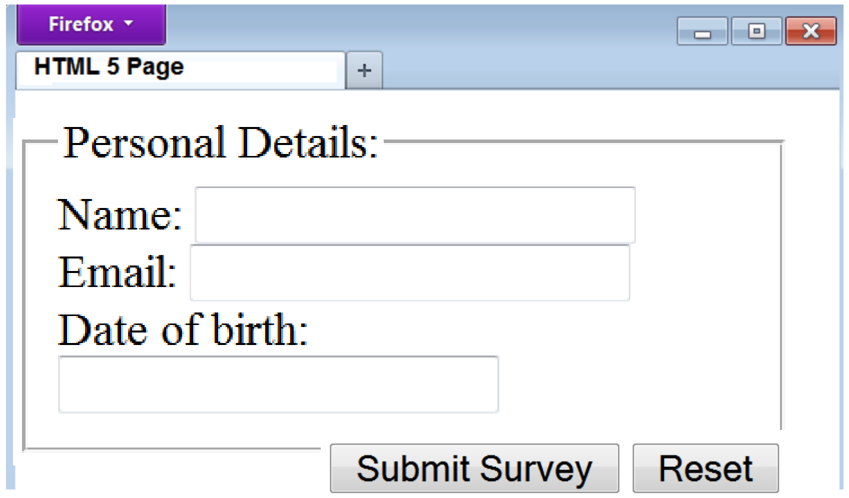
Form element

<form> … </form> provides a mechanism to allow a user to enter information into a web page.

Entered information can be submitted to a server, which it turn can receive the data, process the data and generate a response.

Possible responses may include:

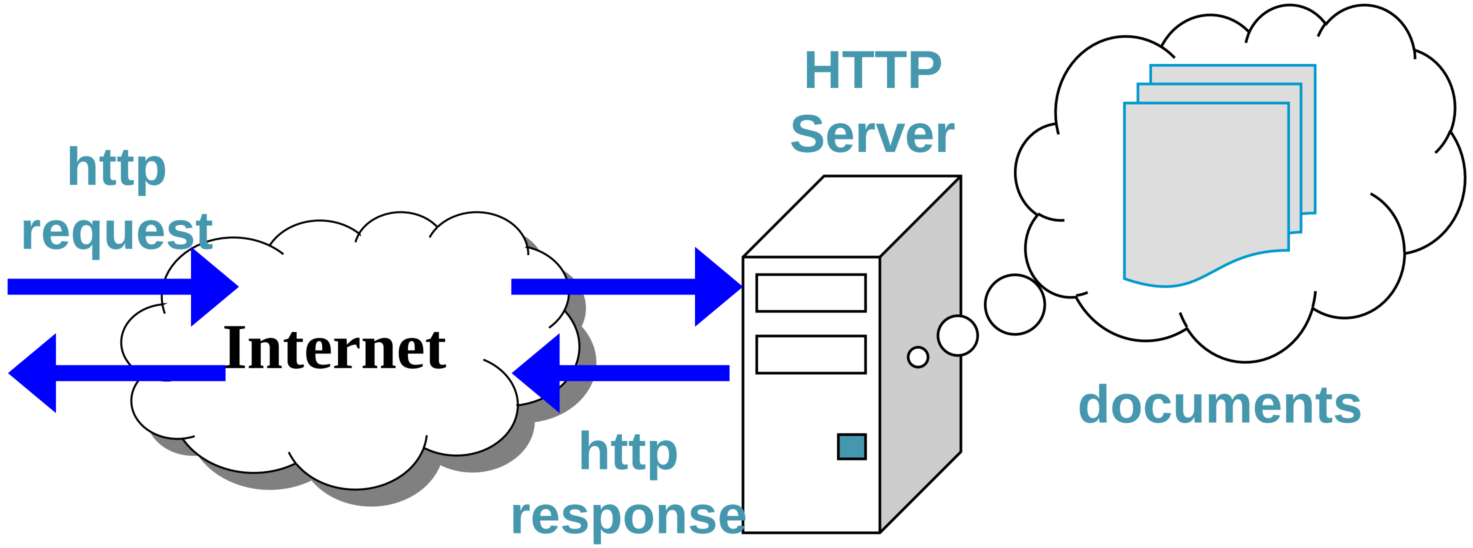
- display information on a web page;
- adding data to a database; or
- sending an email message.



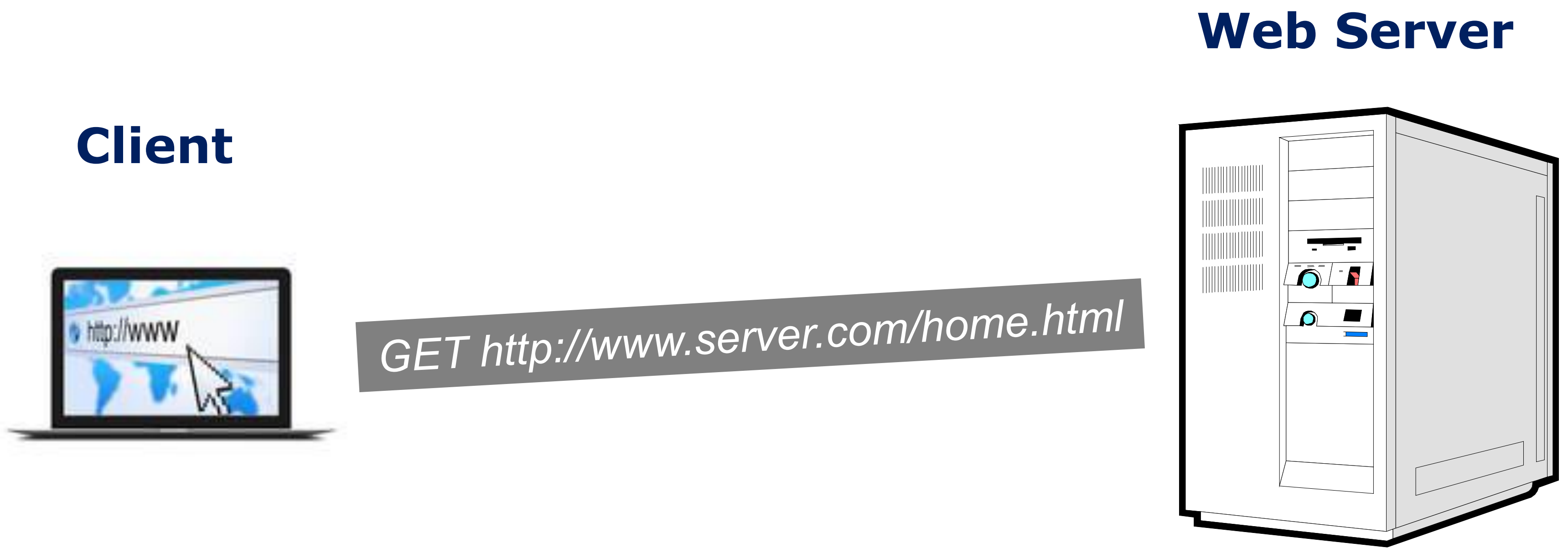
Web Server Features

A Web server is made up of several components:

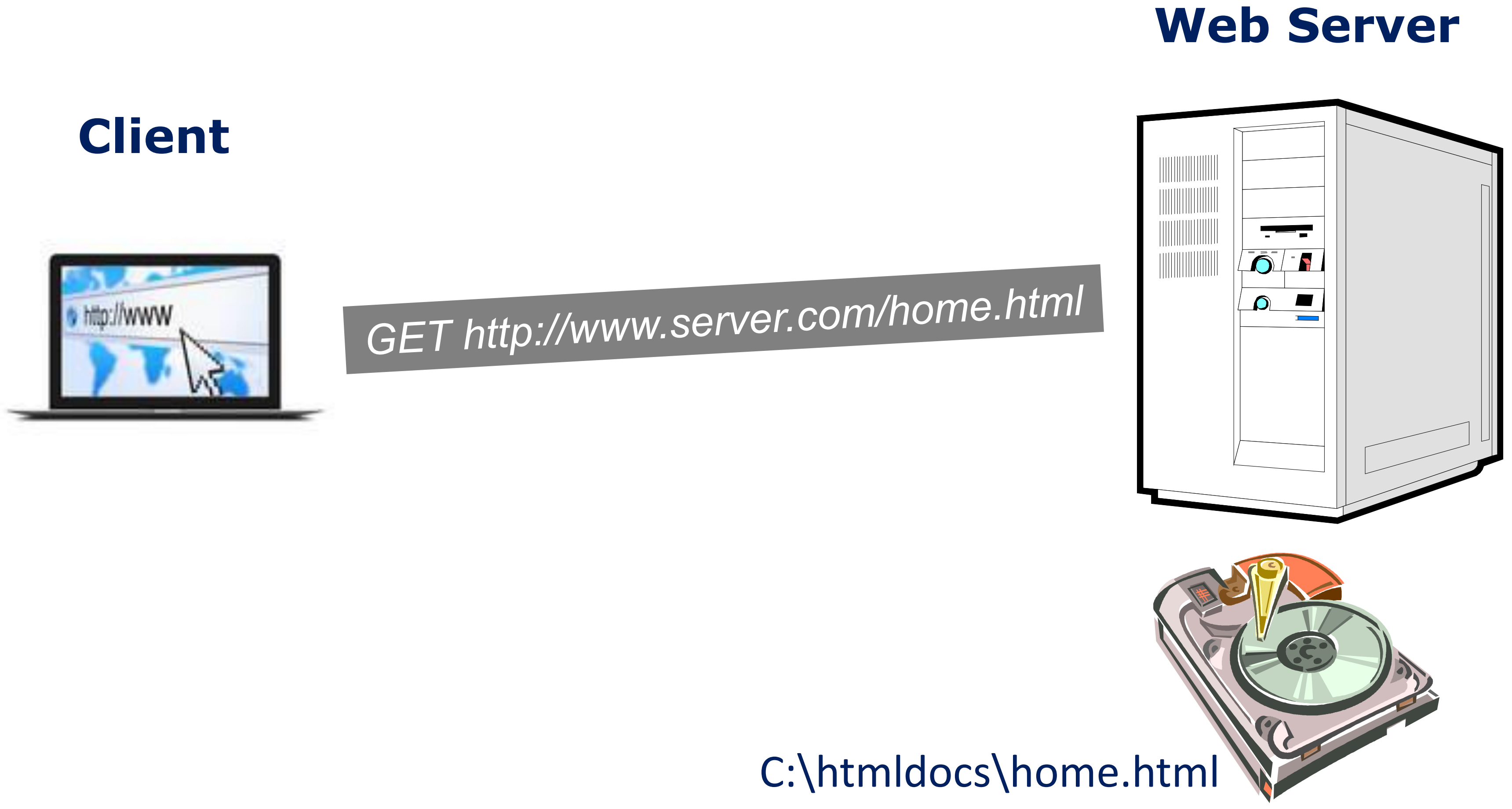
- A **computer** with an **Internet connection** and **operating system**.
  - The server program usually ***runs continuously***.
- **Web server software** to receive and respond to HTTP requests.
  - Handles ***multiple requests***
- **Information**: a collection of documents to be served.
  - Careful ***access control*** to server content should be a feature



The client, using standard WWW technology, sends a request for a document to the Web Server using HTTP. The request takes the form of a URL specification.



The web server locates the file by mapping the URL specified by the client to a file in the local resource base (e.g. hard disk).

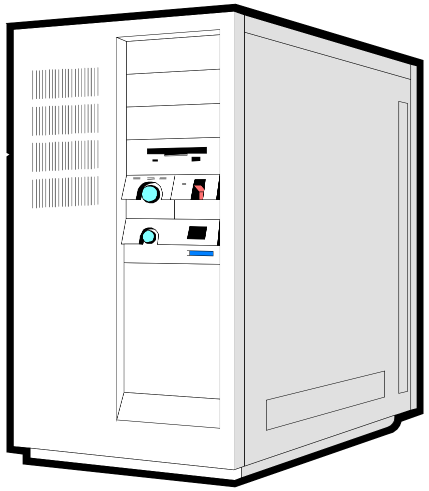


The Web Server sends the file back to the client 'as is' along with specific server generated headers (containing control information for the browser)

Web Server

Client

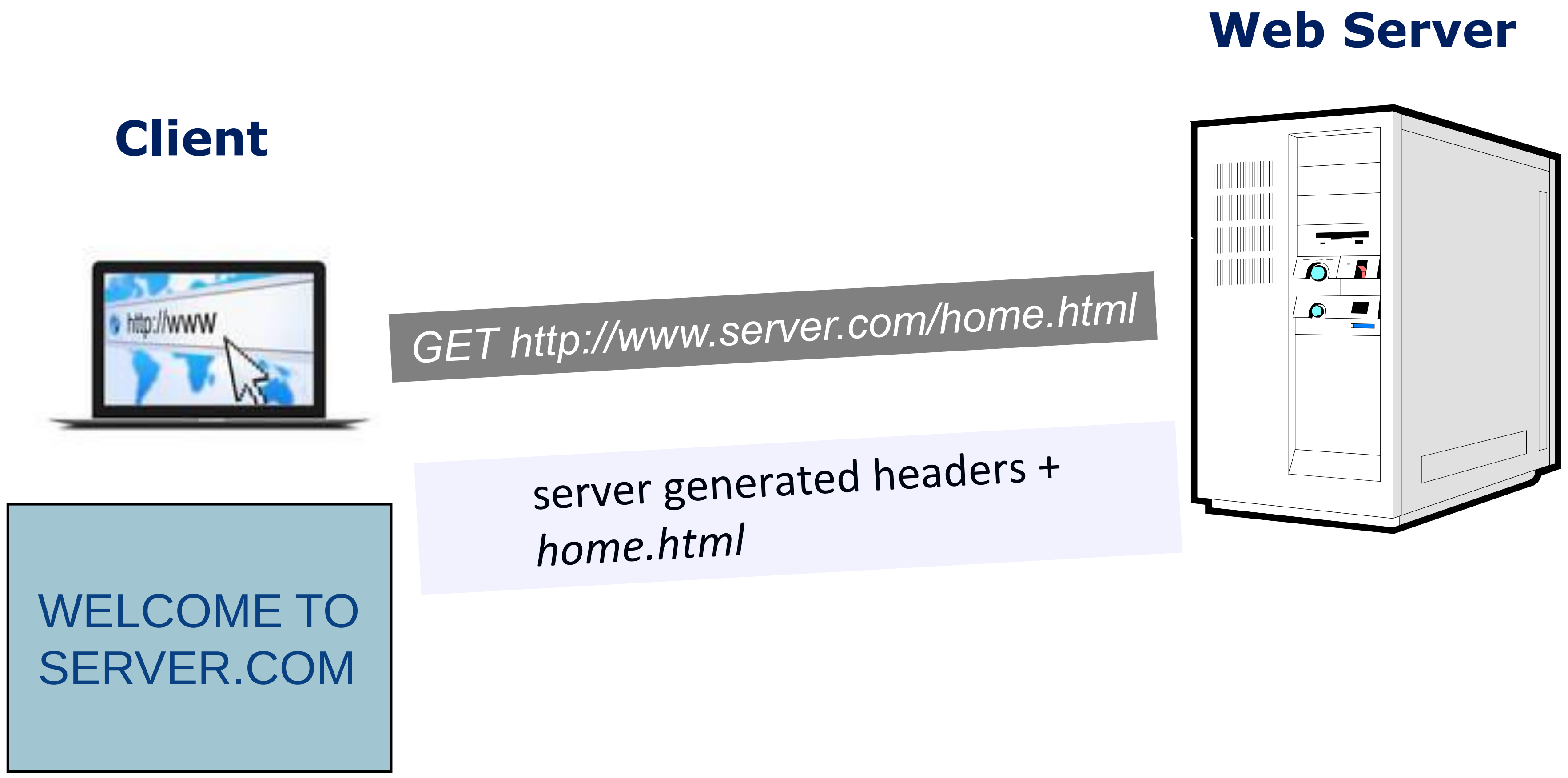
GET http://www.server.com/home.html



server generated headers + *home.html*

```
<html>
<body>
<h1> WELCOME TO SERVER.COM </h1>
</body>
</html>
```

Client web browser interprets and displays (renders) the HTML document.


Web Server
Client
http://www
GET http://www.server.com/home.html
server generated headers + *home.html*
WELCOME TO SERVER.COM

Static vs Dynamic web pages

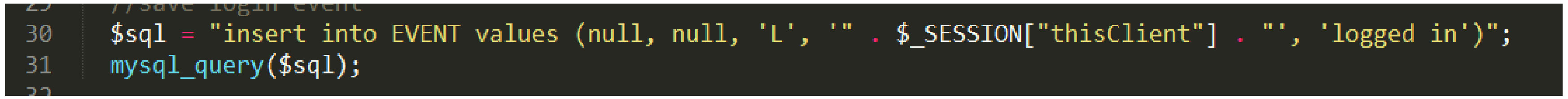
STATIC web page

- the URL identifies a file on the server's file system
- server fetches the file and sends it to the browser
- the file contains HTML
- browser interprets the HTML for display on screen

DYNAMIC web page

- URL identifies a program to be run
- web app runs the program
- program typically retrieves data from database
- elements such as TABLE, LIST are populated with data
  - web app uses LOOPS to fill the contents of TABLEs and LISTs.
  - e.g. SELECT * FROM Product; (returns a set of product entities)
  - FOR p IN ProductList, print a row in HTML table

Problems with old-style web apps



```
//save login event
$sql = "insert into EVENT values (null, null, 'L', '" . $_SESSION["thisClient"] . "', 'logged in')";
mysql_query($sql);
```

Placing “raw” SQL inside PHP/HTML files

- Mixes presentation, business, database logic
- Hard to maintain when things change
- Want separation of concerns, e.g. MVC (model view controller)

Lots of reinvention of wheels

- each developer writes their own solution to common features
- e.g. login security, presentation templates, database access

Increasing variety of clients e.g. phones and tablets

- Manually program for different platforms

=> web application frameworks

- examples: Ruby on Rails, .Net, Symfony, AngularJS, Django

Web Services

The WWW allows humans to access remote databases

Web Services allow *computers* to access remote databases

2 major approaches: SOAP and REST

- Simple Object Access Protocol
- Representational State Transfer

Structured data usually returned in XML or JSON format

REST nouns are resources, addressed via URIs

REST verbs correspond to DML statements

GET (select), POST (insert), PUT (update), DELETE (delete)

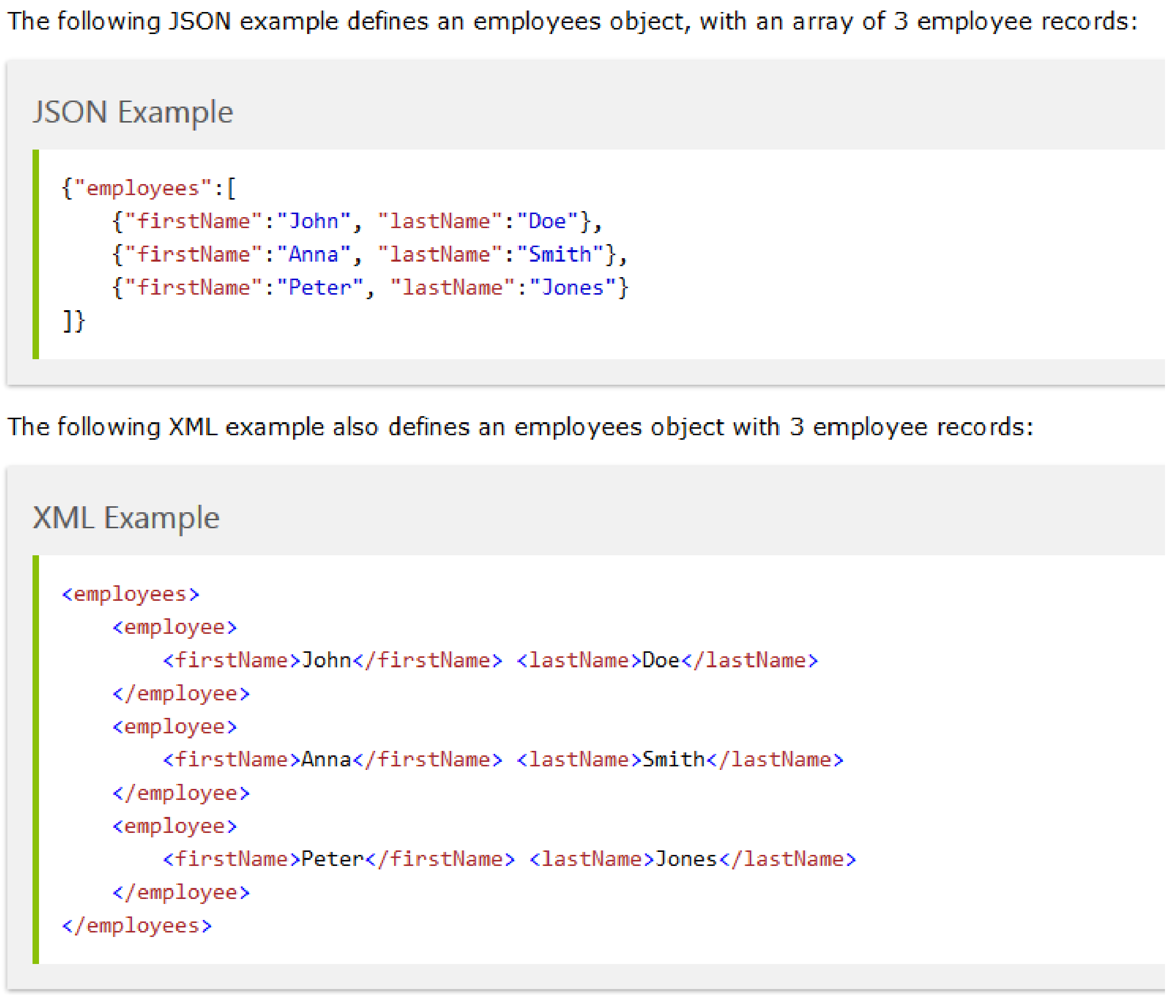
Try this example web service
https://www.googleapis.com/books/v1/volumes?q=quilting

XML and JSON data formats

used by web services for data exchange

- XML
  eXensible Markup Language
- JSON
  JavaScript Object Notation

Source: www.w3school.org



The following JSON example defines an employees object, with an array of 3 employee records:

JSON Example

```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

The following XML example also defines an employees object with 3 employee records:

XML Example

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

What’s examinable

Identify the limitations of SQL

Distribution of Processing Logic

Database Architectures

Web languages

Web architecture

HTML elements

How static and dynamic web pages work (high level)

THE UNIVERSITY OF MELBOURNE

Thank you