Web Programming

Lecturer: Toby Daniel

ICS313

Course Description

- An introductory study of concepts and technologies of World Wide Web and Internet programming. It covers the features of HTML, CSS, Javascript, AJAX, PHP, and Ruby on Rails including basic database management and usage.

ICS313

Prerequisite: ICS 110 Computer Programming I

Assessment:

- Lab Assignments 10%
- Individual Project 20%
- Mid-term exam 30%
- Final exam 40%

ICS313 Topics

- HTML, Cascading Style Sheets
- Introduction to Javascript
- Web database concepts - MySQL
- XML, AJAX, AJAX frameworks
- Introduction to PHP, PHP and DBMS
- Final Project introduction
- Introduction to Ruby, Introduction to Rails, Ruby on Rails
- MVC and frameworks
- Final Project and demonstration

Intro to Web Programming

Web programming is used to build:

- web pages
- web sites
- web applications

Intro to Web Programming

Web Page

- A document containing text and graphics created with HTML that is part of a group of hypertext documents or resources that can be accessed through a web browser on the internet.

Intro to Web Programming

Web Site

- A collection of related web pages found at a single address.
- A URL serves as the top-level address of a Web site and points to that Web site's home page. That page serves as a reference point, containing pointers to additional HTML pages or links to other Web sites.

Intro to Web Programming

Web Application

- A web application is a set of web pages that are generated in response to user requests.
- The Internet has many different types of web applications, such as
  - search engines
  - online stores
  - auctions
  - news sites
  - games, etc

Intro to Web Programming

- Hypertext Markup Language, or HTML, is the language that the web browser converts into the web pages of a web application.
- Hypertext Transfer Protocol, or HTTP, is the protocol that web browsers and web servers use to communicate.

Intro to Web Programming

Types of web pages:

- static web page
- dynamic web page
  - client-side scripting
  - server-side scripting

Intro to Web Programming

Static Web Pages

- A static web page is an HTML document that's stored in a file and does not change in response to user input. Static web pages have a filename with an extension of .htm or .html

Intro to Web Programming

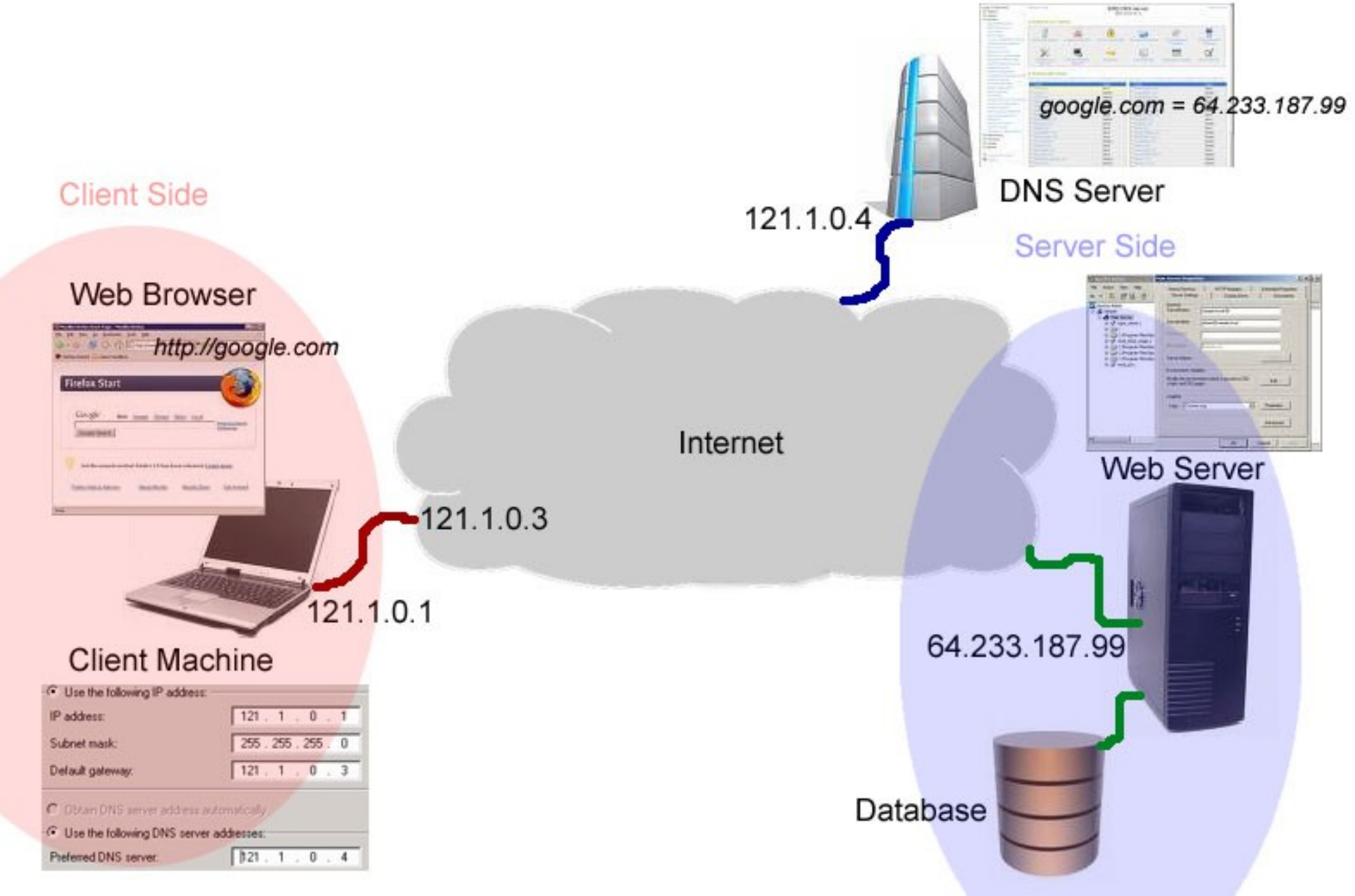
Dynamic Web Pages

- A dynamic web page is an HTML document that's generated by a web application. Often, the web page changes according to parameters that are sent to the web application by the web browser.

How static web pages work

- The process begins when a web browser requests a web page - by entering a web address into the browser's Address box or clicking a link.
- The web browser sends an HTTP request to the web site's server.
- When the web server receives an HTTP request it retrieves the requested HTML file from disk and sends the file back to the browser - an HTTP response.
- The browser receives the HTTP response, it formats and displays the HTML document.

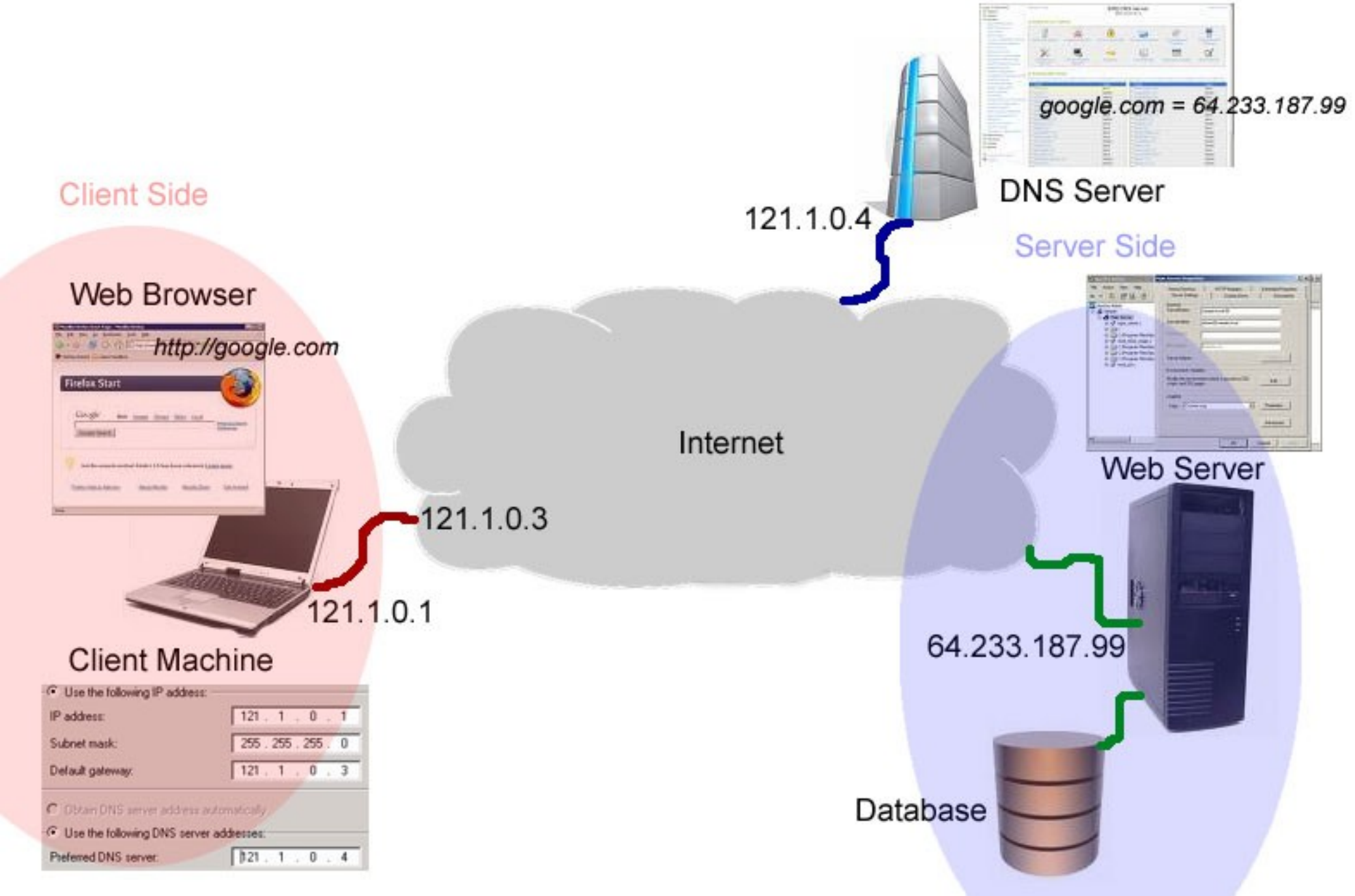
How the web works!



How dynamic web pages work

- A dynamic web page changes based on the parameters that are sent to the web application from another page.
- The web application generates the HTML for a new web page and sends it back to the browser.
  - example: search engine

How the web works!



Server-side scripting:

- PHP Hypertext Preprocessor (PHP)
- Java server pages (JSP)
- Active server pages (ASP)
- Ruby on Rails
- ColdFusion
- Escapade (ESP)
- Perl

Client-side scripting:

- JavaScript
- VBScript

Differences between client and server-side:

- Client-side scripts have greater access to the information and functions available on the user's browser.
- Server-side scripts have greater access to the information and functions available on the server.
- Server-side scripts require that an interpreter is installed on the server.
- Client-side scripts do not require additional software on the server but they require that the user's web browser understands the scripting language (browser compatibility problems)

Benefits of Web Technologies

Q. What are the benefits of using web technologies to build applications?

(compared to traditional client/server applications or desktop applications)

Benefits of Web Technologies

- Less Expensive Infrastructure:
  - Since every desktop can be equipped with a free browser, the users only need inexpensive, low maintenance computers that run the browser software.
- Rapid Software Distribution:
  - Thanks to the web, there is no need to distribute new software to any of the client browsers. Once programs are updated on the server, users can instantly see the changes the next time they load the page into their browser. Troubleshooting is also improved since most issues can be investigated and resolved on the server.

Benefits of Web Technologies

- Low Network Usage:
  - In a web environment, the client application (browser) doesn't access the database over the network to retrieve or update the data. The browser only sends instructions to the server with as much information as necessary to execute the server program. The server software is responsible for communicating with and retrieving any content from the database and composing the page to be sent back to the client; there is no need to exchange large amounts of data.
- Platform Independence:
  - Most web applications can be accessed through a variety of web browsers like Internet Explorer, Firefox, etc. Web browsers exist for virtually any operating system. Since they receive only the HTML document it is irrelevant if the document is being viewed on Windows, Linux/UNIX, Mac, or any other system. Web applications are accessible through wireless internet devices .

Challenges of Web Technologies

Q. What are the challenges of using web technologies to build applications?

(compared to traditional client/server applications or desktop applications)

Challenges of Web Technologies

- Limited Interactivity:
  - Interactivity is limited since a web page cannot communicate with the server in real-time except when the page is submitted to the server by the user. For example, if a program needs to populate a particular list box field, it needs to refresh the whole page when the user reaches a particular field on the screen.
- Lack of Skills:
  - The web is still foreign to many developers who have been busy developing mainframe, client/server, and standalone applications. These developers are often unfamiliar with web technologies, web programming, and web scalability issues.

Challenges of Web Technologies

- Security Risks:
  - Utilizing the web exposes the company's network to new security risks and programming vulnerabilities that can expose private data to the public.
- Browser Compatibility:
  - Unfortunately, the competition between web browsers has resulted in many differences in supported standards, thus creating confusion between users, and more importantly, compatibility issues with web applications. You have to test your web application with several browser versions, on several operating systems.

Step 1.

Create a web development machine!

- A web development machine is a computer that has both the client side and server side software on one machine.
  - this means that while you are programming you don't need an Internet connection or an external web server to test your projects.