Web Programming

Dr Hossein Nevisi

Department of Computer Science
Loughborough University

About Module

- Lecturer:
  - Dr Hossein Nevisi
  - E-mail: H.Nevisi@lboro.ac.uk
  - Office: N.3.13

- Technical Tutor: Dr Gary Storey (email: G.Storey@lboro.ac.uk)

- Lectures: Mondays, 4pm-6pm; Thursdays 2pm-4pm (including workshop)

- Lab Sessions:
  - Starts from Week 1
  - Tuesdays, 9am – 11am (for Group 1) and 11am-1pm (for Group 2)

- Module Credit:
  - **20** credit (double module)
  - Twice material, twice as much work
  - Teach yourself through practice

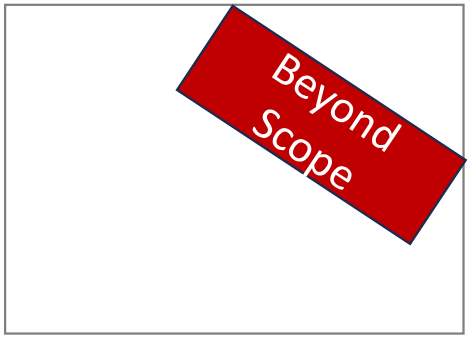
How to ask questions?

- Ask your questions during the lab sessions
- Drop me email about the topic or concept that you need more explanation on
  - will be discussed in the beginning of the next lecture
- During the lectures, you can ask questions about the topic that I am talking about
- Arrange a meeting

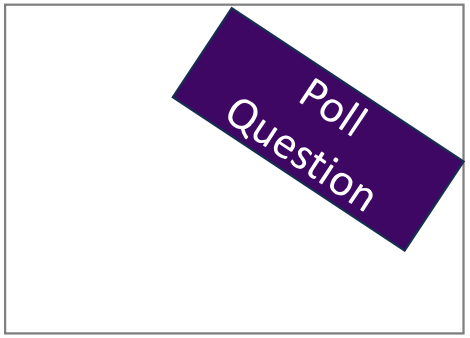
- Please don’t send me your code to debug

Slide Labelling

Material for independent learning. Study this at your own convenience and explore the content to enhance your understanding of the subject matter



Material that goes beyond the standard module content, tailored for those with a deeper interest or seeking additional challenges



Vevox multiple choice question

Assessment

- Coursework (100%)
  - Will be published on Learn before Easter break

- How to prepare for the coursework?
  - Active engagement in lectures, labs, and workshops
  - Self-learning through practice

Workshop

- Objective:
  - Enhance learning outcomes
  - Evaluate understanding of previously covered material
  - Proactive preparation for upcoming topics of the next lecture
  - Promote teamwork, communication, and problem-solving skills

- Tasks Overview
  - Focused on group collaboration
  - Designed to support concepts discussed in lectures and prepare for upcoming topics
  - More challenging than typical lab exercises
  - Showcasing real-world scenarios

- Feedback
  - Regular feedback from both peers and instructors

Workshop – My Expectations

- Bring your laptops (with sufficient charge)
- Quickly form groups before the workshop begins
- Limit group size to a maximum of 3 members (as much as possible)
- Ensure that individuals who have experience in web development do not sit in the same group
- Ensure you sit where allows instructors to approach you and monitor your progress effectively
- Review the lecture materials before attending
- Participate actively in group discussions and activities
- Establish a collaborative environment by sharing insights and ideas

Module Aims

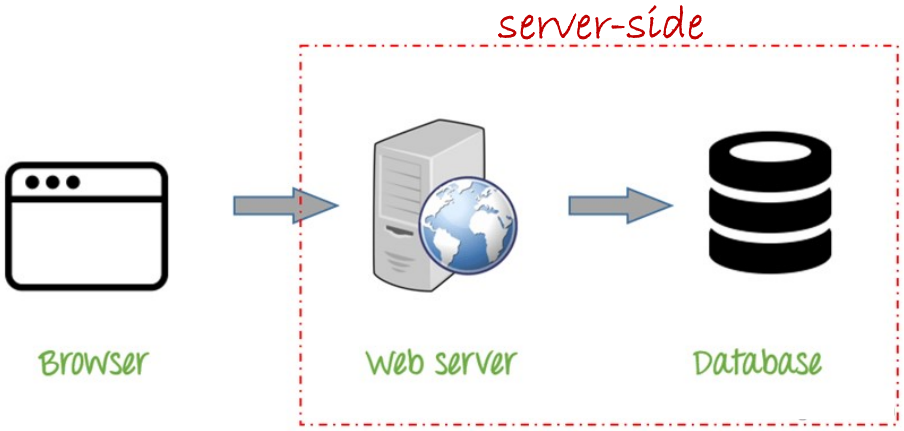
To acquire up-to-date knowledge and skills in

- developing dynamic web applications using client side (frontend) and server side (backend) scripting
- Html, CSS and JavaScript
- JQUERY and REACT (popular JavaScript libraries)
- PHP
- MySQL database management system and Structured Query Language (SQL)
- AJAX (**A**synchronous **J**avaScript **A**nd **X**ML) using XMLHttpRequest and Fetch API

Module Content

- Develop the client-side (frontend) of a web application:
  - **HTML** (Hypertext Markup Language): is the standard markup language for creating Web pages (a markup language is a system for annotating a document)
  - **CSS** (Cascading Style Sheets): is a style sheet language used for describing the presentation of a document written in a markup language
  - **JavaScript**: is the most popular programming language of the Web. It controls the behaviour of different elements and it makes the site interactive for the user
  - **jQuery** and React (JavaScript libraries)

- Develop the server-side (backend) of a web application:
  - **PHP**: is a general-purpose scripting language especially suited for web development (server-side scripting)
  - **MySQL**: is a database management system and Structured Query Language (SQL)
  - **Fetch API and XMLHttpRequest (AJAX)**: is a set of web development techniques (using JavaScript) that enables web applications to send and retrieve data to/from a server asynchronously without interfering with the display and behaviour of the existing page



Client-Side (front-end) vs Server-Side (back-end)

:: Video ::

Required Software

- A **text editor** is enough to write HTML, CSS, JavaScript, and PHP
  - **Notepad++**
  - **Emacs**
  - **vi / Vim**
  - **BBEdit**
  - **Visual Studio Code**
  - ...
- Online (HTML/CSS/JS) editors can be used as well for quick testing and previewing:

  https://www.w3schools.com/tryit/
  https://jsfiddle.net/
  https://www.codeply.com/

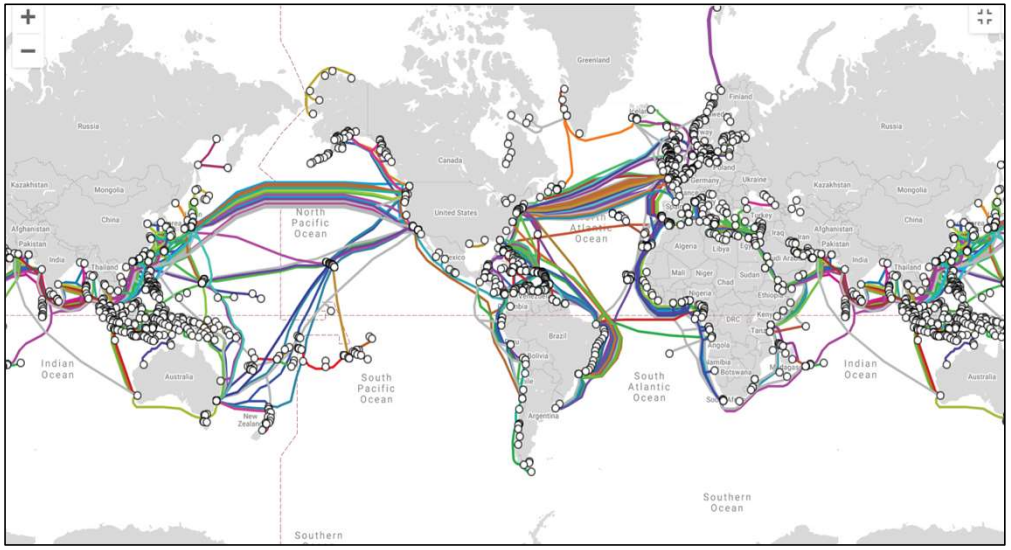
- **Web Browser**: Chrome
- **Web Server (web host)**: sci-project.lboro.ac.uk
- **Off-Campus Access (VPN)**: **Cisco AnyConnect**
- **File Transfer Application**: Secure file-transfer between client and server
  - WinSCP
  - Cyberduck (for MacOS)
- **MySQL administration tool**: phpMyAdmin (available on sci-project server and sci-mysql)

Online Resources

- https://www.w3schools.com/html/
- https://developer.mozilla.org/en-US/docs/Web/HTML
- https://www.w3schools.com/css/
- https://developer.mozilla.org/en-US/docs/Web/CSS
- https://www.w3schools.com/js/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://www.w3schools.com/jquery/
- https://jquery.com/
- https://reactjs.org/tutorial/tutorial.html
- https://www.w3schools.com/REACT/DEFAULT.ASP
- https://www.w3schools.com/php/
- https://www.php.net/
- https://dev.mysql.com/doc/
- https://www.w3schools.com/sql/
- https://api.jquery.com/jquery.ajax/
- https://www.w3schools.com/xml/ajax_intro.asp
- https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

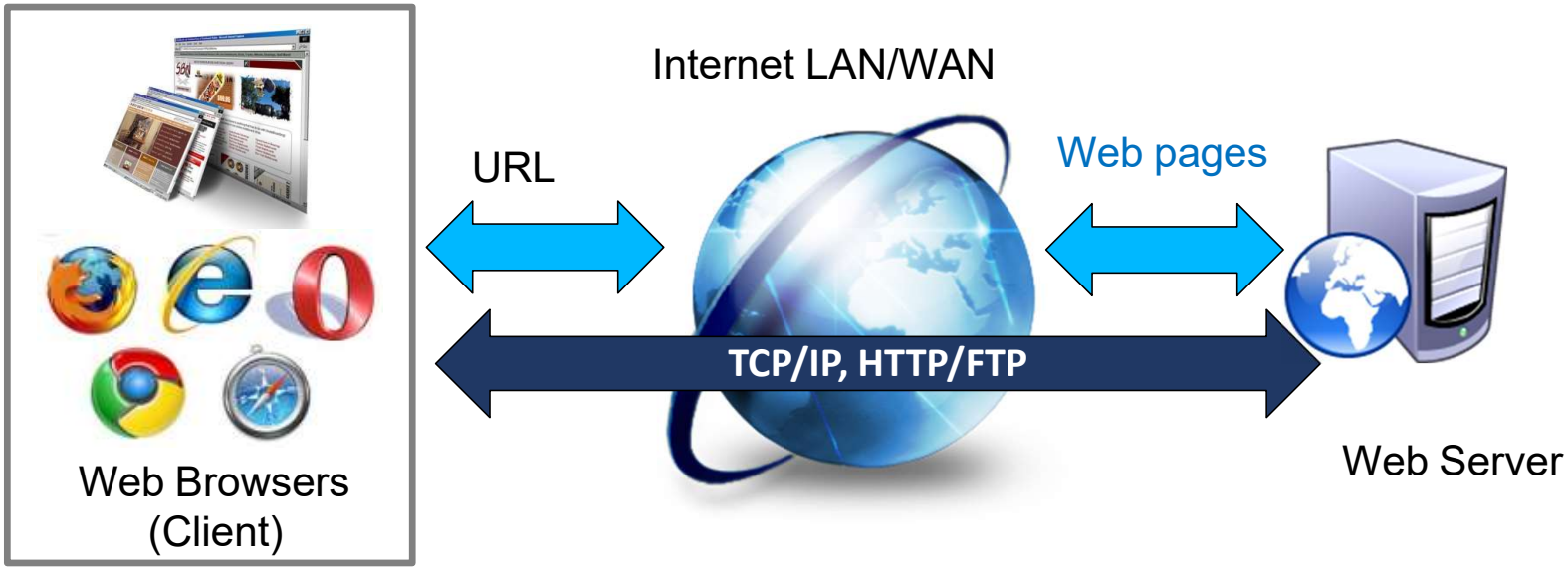
World Wide Web (WWW)

- What is Internet?
  - Internet is the global system of interconnected computer networks that uses the Internet protocol suite (TCP/IP) to communicate between networks and devices. These are linked by a broad array of electronic, wireless, and optical networking technologies (see a visual representation of the network of undersea cables that form a significant part of the global Internet infrastructure: https://www.submarinecablemap.com)



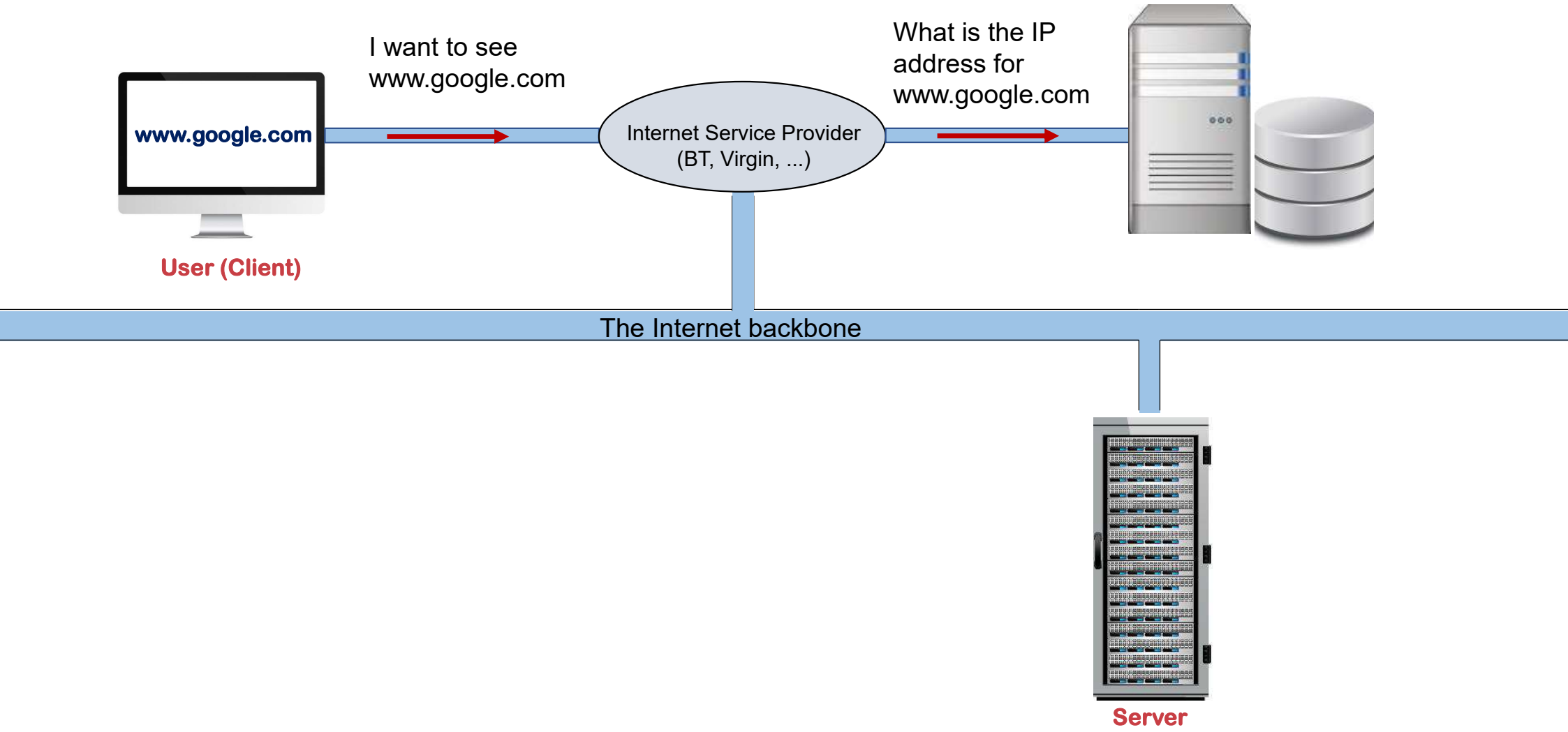
- The World Wide Web (WWW), commonly known as the Web, is a hypertext information system where documents and other web resources are identified by URLs (Uniform Resource Locators, such as https://example.com/) and are accessible over the Internet

Client – Server Architecture

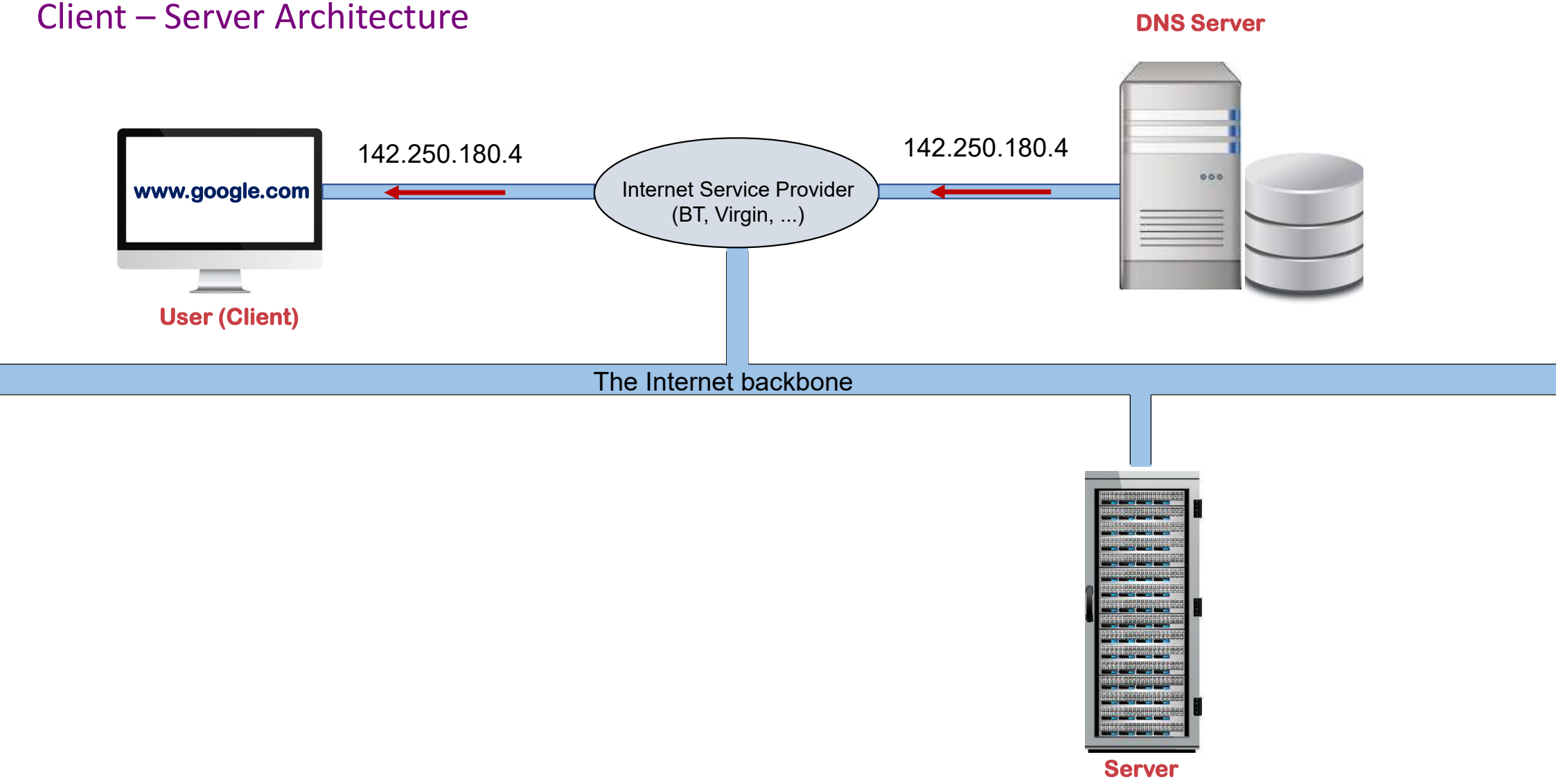


- Client sends URL via HTTP (Hypertext Transfer Protocol) to servers
- Server responds with the requested web page (or error message)
- Client web browser renders returned web page
- The entire system runs on networking protocols (e.g. TCP/IP) over Internet

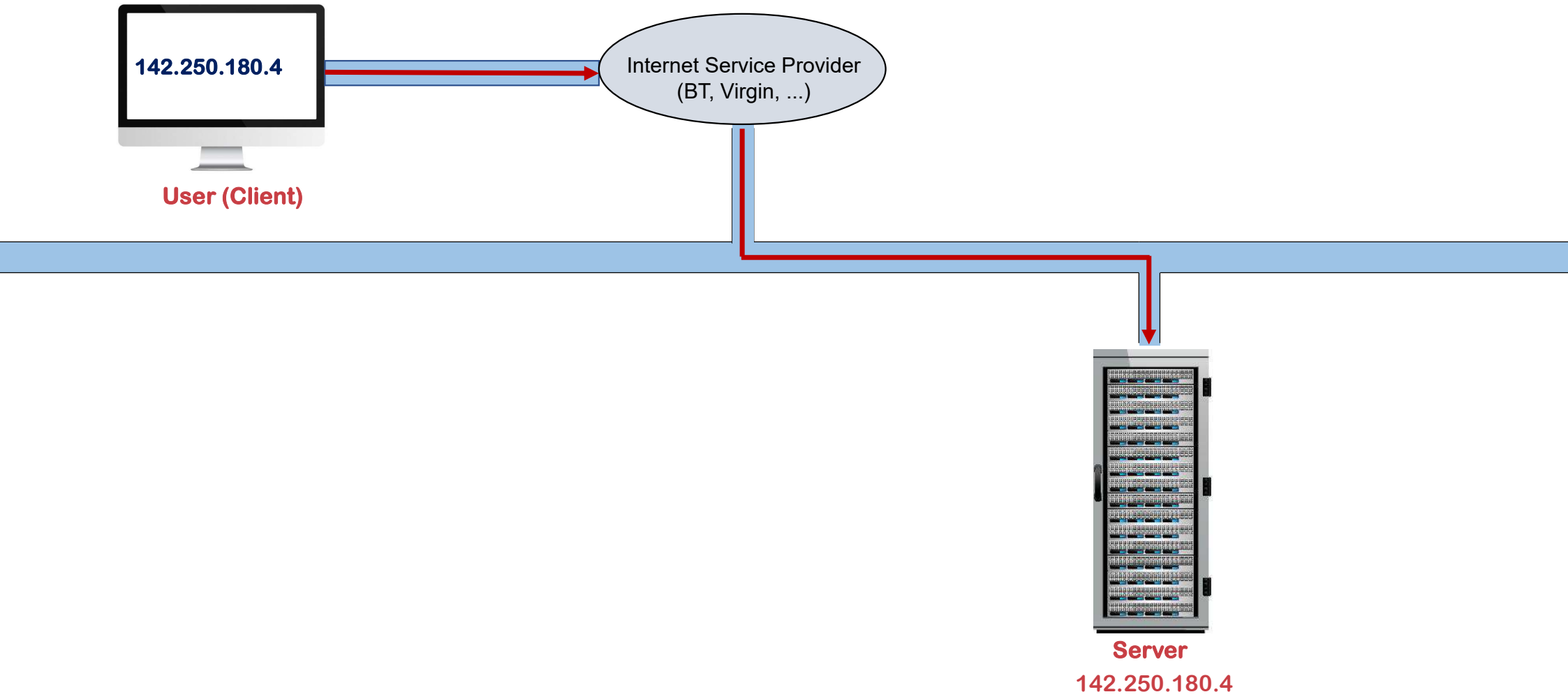
Client – Server Architecture



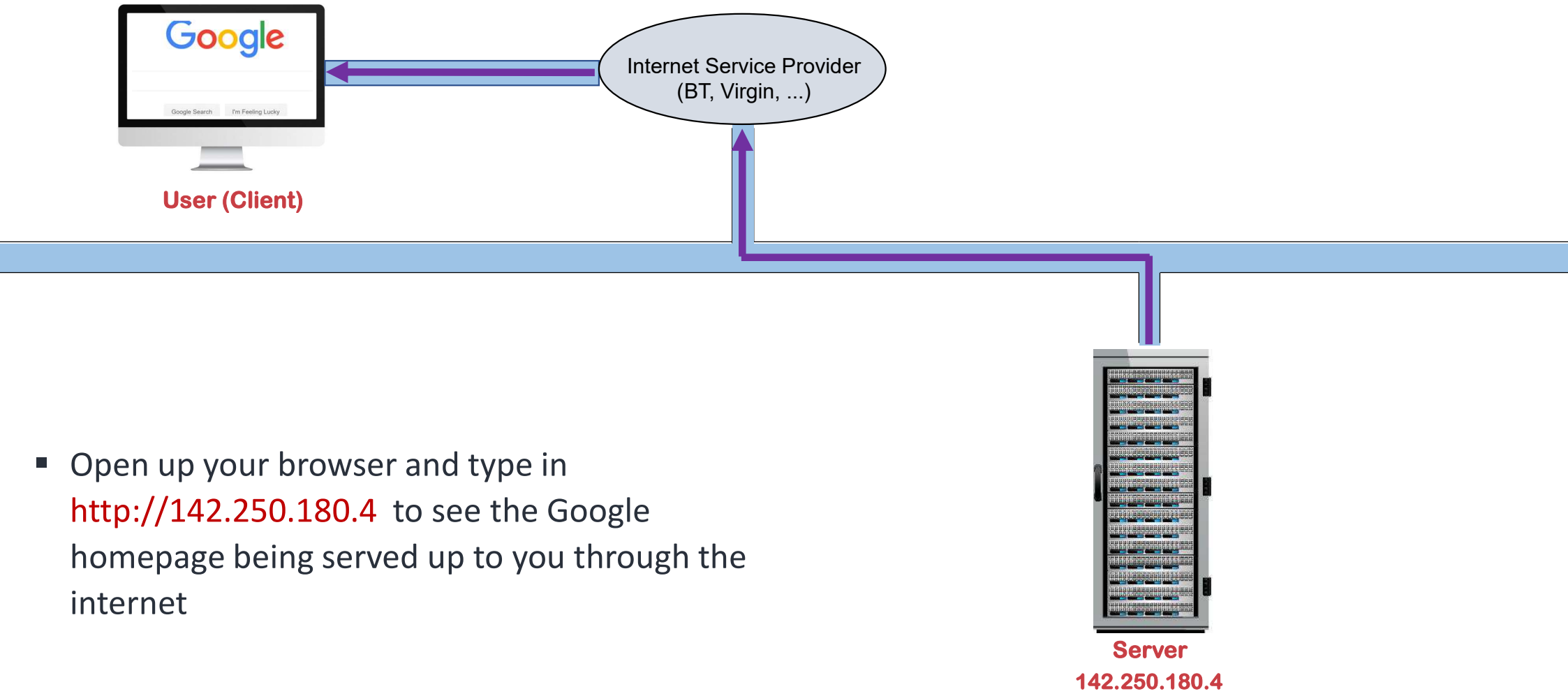
Client – Server Architecture



Client – Server Architecture



Client – Server Architecture

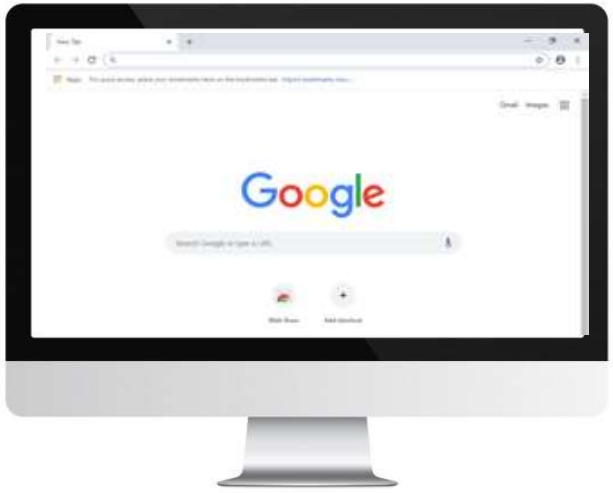


- Open up your browser and type in http://142.250.180.4 to see the Google homepage being served up to you through the internet

Website Front-end

- HTML (Hypertext Markup Language)
- CSS (Cascading Style Sheet)
- JS (Java Script)

Server



User (Client)

Server

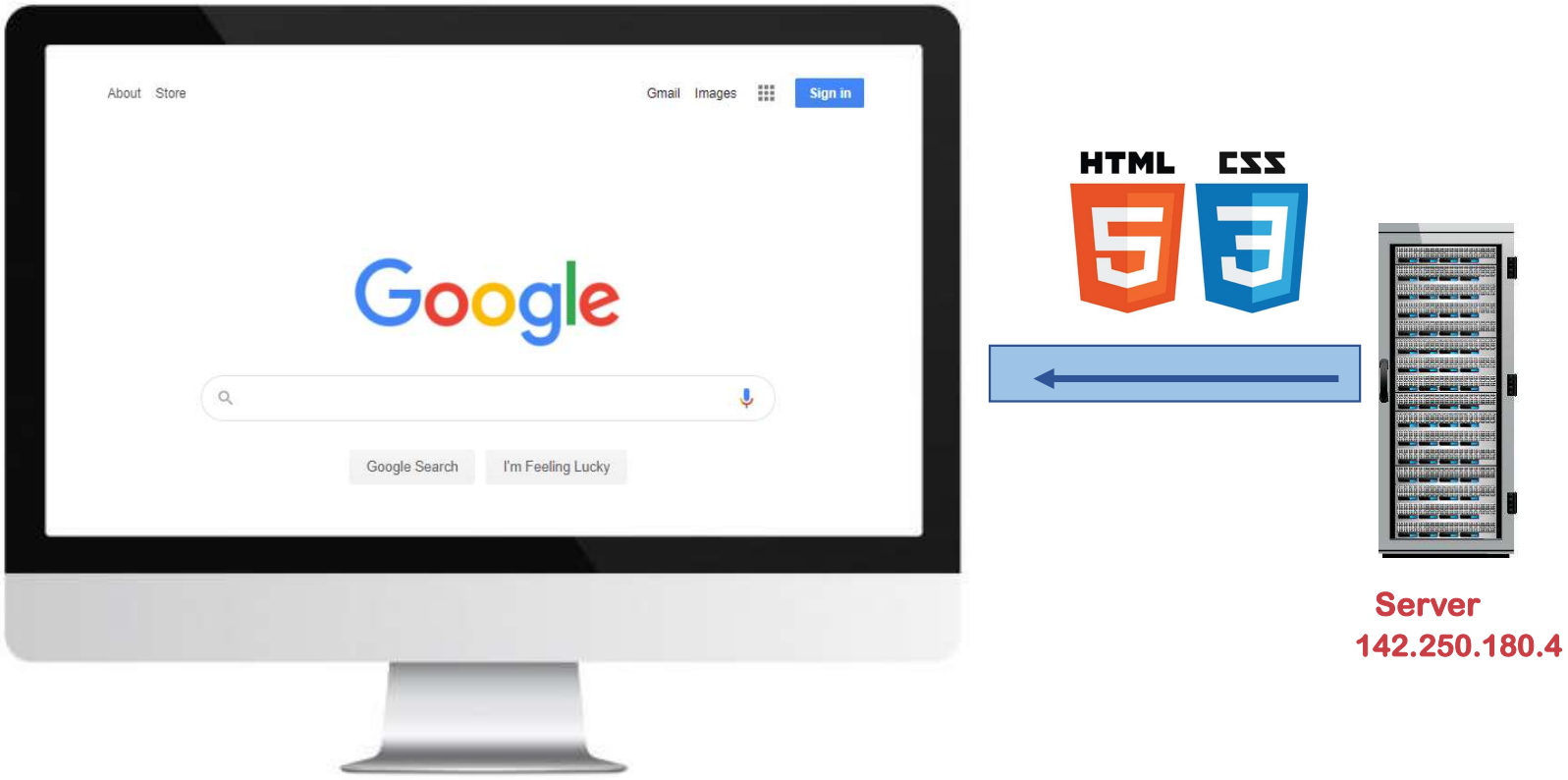
142.250.180.4

HTML (Hypertext Markup Language)

- The skeleton of a webpage and is the very first thing being loaded in the web browser.
- Here the developer defines what elements should the website have: Buttons, Images, Text, Links, etc.

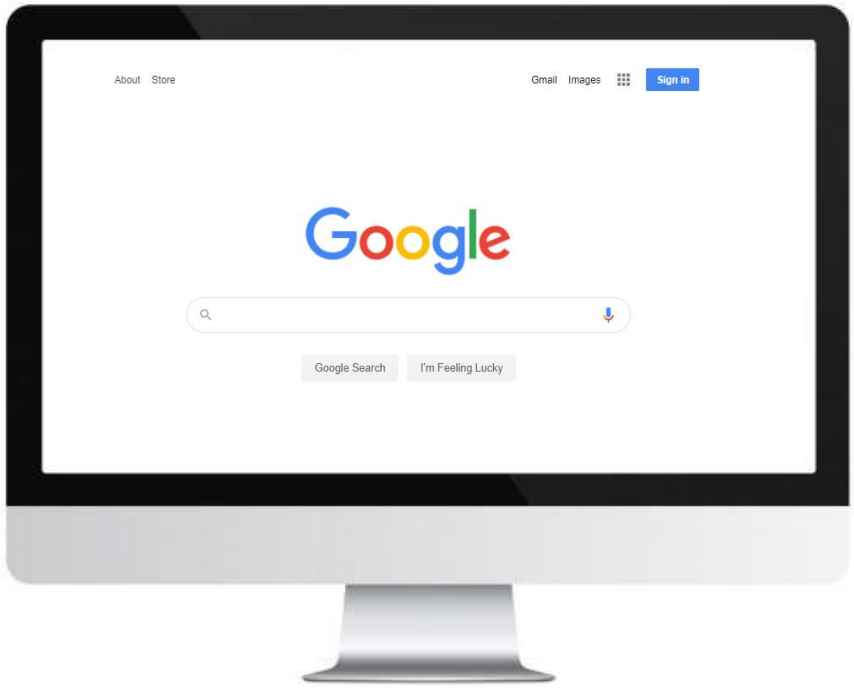
CSS (Cascading Style Sheet)

- A style sheet language used for describing the presentation of a document written in a markup language such as HTML
- Including colours, layout, and fonts



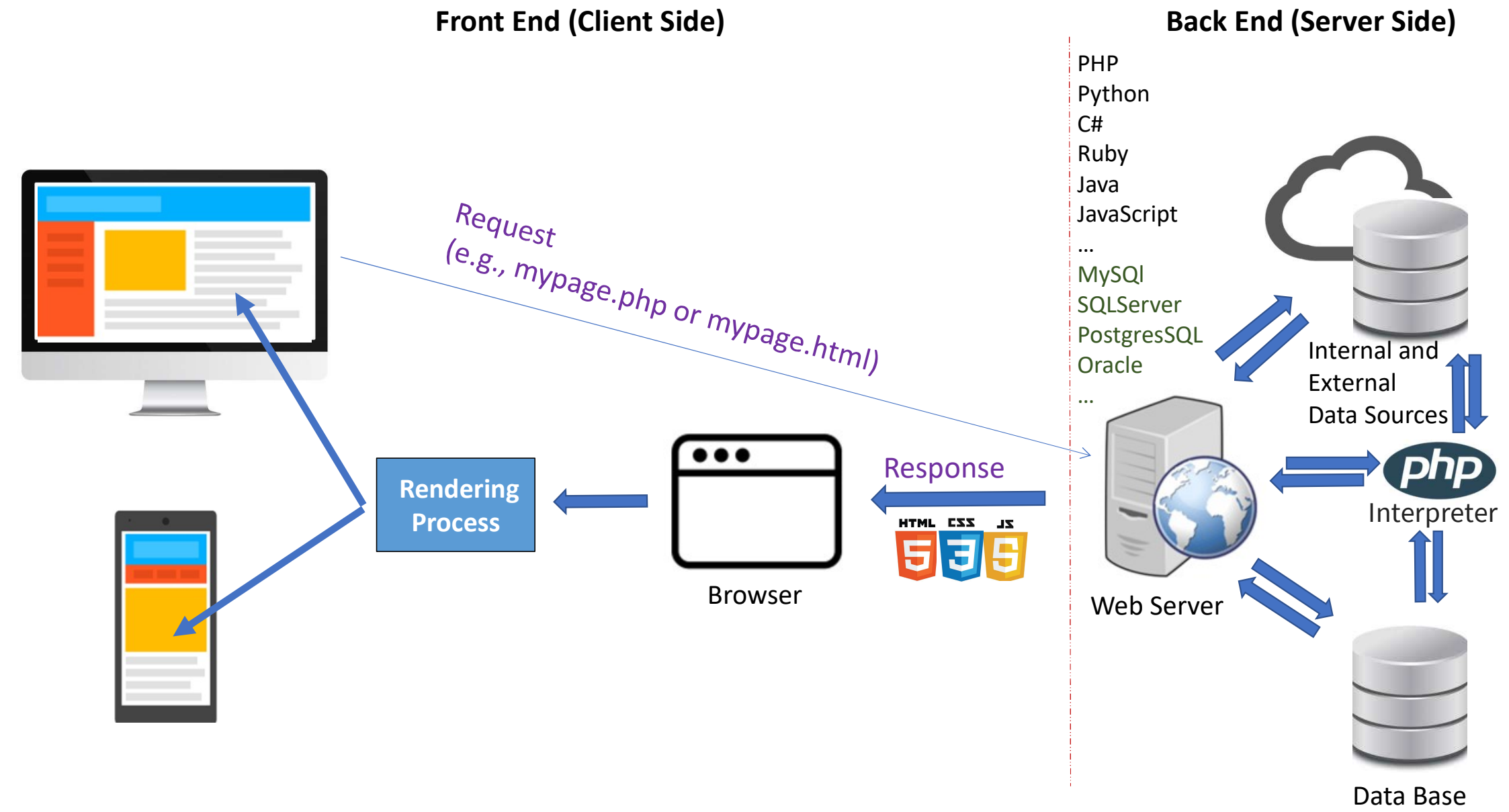
JS (Java Script)

- JS is used to control the behaviour of different elements and it makes the site interactive for the user
- It is a real programming language where you have functions, variables, arrays and many other programming tools to create “logic” within your site



User (Client)

Server
142.250.180.4


Front End (Client Side)
Back End (Server Side)
PHP
Python
C#
Ruby
Java
JavaScript
...
MySQl
SQLServer
PostgresSQL
Oracle
...
Request
(e.g., mypage.php or mypage.html)
Internal and
External
Data Sources
Rendering
Process
Browser
Response
HTML
CSS
JS
Web Server
php
Interpreter
Data Base