
Web Programming

— Course Introduction —

Mahsa Saeidi

The World Wide Web

- Original idea (Tim Berners-Lee, 1989)
 - Public information sharing on Internet
 - Hypertext
 - Documents are text which can be displayed/converted to desired output
 - Documents can be linked to each others:
- WWW: A system of interlinked hypertext
 - Now, much more complex/interesting applications



How Does WWW Work?

- Client-Server mechanism
- Web servers: Process client's requests
 - File (text, image, video, ...) retrieval requests
 - Computation/Processing (DB lookup, transaction, ...) requests
- Web clients: Send the requests
 - Browser: Interacts with client, Requests for server, Processes and displays response (rendering)
 - Other applications
 - Search engines crawlers
 - Use server as a processing element (distributed computing)
 - ...



WWW: From Old to Now

- Static Web Pages
- Client requests a document from server
 - A communication protocol: HTTP
- How to display the document in browser?
 - Document structure definition language: HTML
 - Representation of document: CSS
 - Later, very later, some advance features: HTML5



WWW: From Old to Now

- Needs to interact with user (e.g., event handling in web pages)
 - A programming language in browser: JavaScript
- Dynamic data from server (e.g., search result)
 - A programming language in server: JavaScript (Node.js)
- Interactive & Dynamic web page
 - A communication mechanism between web page and server: JavaScript & JSON



Let's Go Through a Web Page

What makes a web page work?

What are the components of a web page?

Code Languages

- **HTML, CSS, and JavaScript (JS).**
 - These are the languages that make up the code that's run by the user's browser:
 - HTML specifies the content and structure of the page.
 - CSS specifies the layout and style of the page.
 - JS specifies the behavior, logic and interactivity of the page.

Page Components

- A web page itself is comprised of many individual components, e.g. the navbar, the search box, the advanced search panel, photo lists, photos, etc.
- The code for each one of these components (content, design, and logic) is factored in such a way as to encapsulate the code and even to make it reusable.

UI/UX

- **Much attention to design and user experience.**
 - A website may have a team of visual designers who develop the complete look and feel of the site, from the icons to the interactions.
 - User studies and testing also play a large role in the site's design.

Stored Data

- A web may have lots of kinds of data (photos, metadata, user data, etc.). Each of these is stored in a particular way on the back end.
- For example in Flickr, photos are stored as files on huge file servers, while photo metadata and user data are stored in a giant DB.

Serving infrastructure.

- A website may be served by many, many machines. In fact, different parts of the site are served by different machines.
- Each time a user requests a page, the serving infrastructure needs to figure out from which machine to handle the request.
 - This involves a combination of routing and load balancing.

Server Logic

- Every time a user makes a request, the serving stack needs to make database or even third-party calls to compile the data from which to create the page that's returned to the user.

User-specific Data

- Every user may have his or her own private data, and they must authenticate themselves to be able to access it.
 - This involves lots of machinery, e.g. logins, cookies, and user sessions.

Application Data Models

- Data from the the website's backend must be modeled within the application of the website so it can be accessed and used.
 - A website may have separate data models for different types of data including search results, photo data, user data, etc.
 - For example, Flickr website follows an MVC architecture.

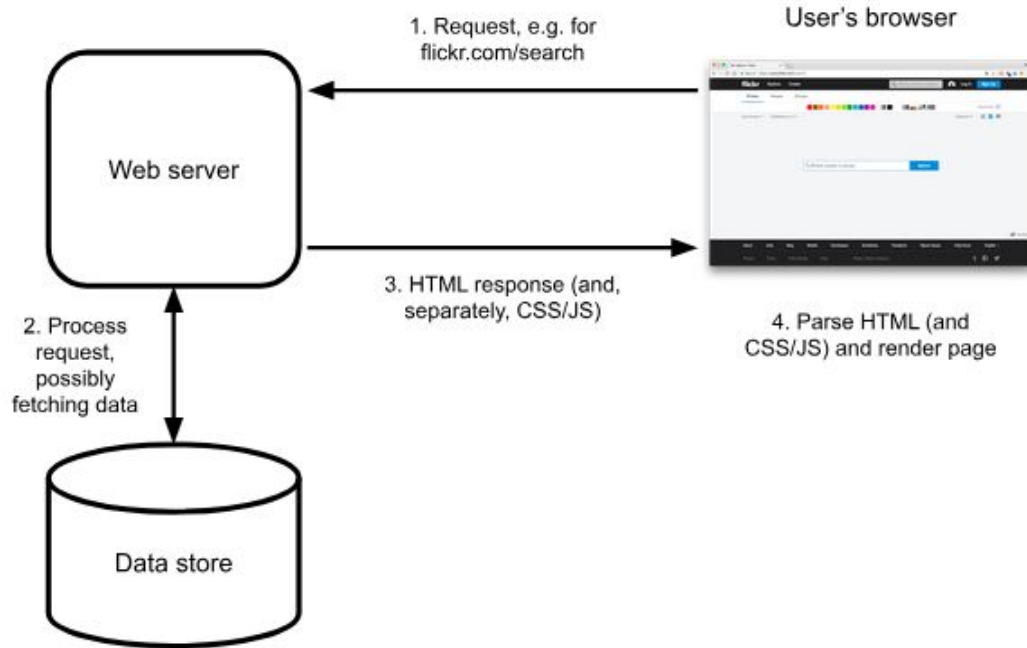
Security, on many levels.

- A site like Flickr has many ways of being attacked and must defend against all of them:
 - Preventing users from injecting malicious content into photo titles, descriptions, comments, etc.
 - E.g. HTML, scripts, database commands.
 - Preventing hacking and vulnerabilities on the back end.
 - E.g. <https://imageragick.com/>
 - Protecting against API misuse, e.g. a DoS attack.

Website Architecture

- Client-side code that runs on the user's browser.
- Server-side code that runs on webs servers.
- One or more data stores, where all data including photos, users data, etc. are stored.

Web Page Viewing Process



What Do We Study in This Course?

- **Foundations of web app architecture.**
- **Building a complete web application that uses the following things:**
 - On the client-side:
 - HTML and CSS to define a page's content and how it looks.
 - Client-side JS to provide interactivity to a page.
 - On the server-side:
 - Server-side JS (Node.js) to implement the server logic.
 - Server-side templating to allow us to dynamically generate content for our pages.
 - Server-side routing to aid in dynamic page creation and to help provide an API our client-side code can communicate with.
 - Data storage:
 - Mongo DB to store our application data.
- **In addition to all this stuff, we'll also learn how to use Git and GitHub to manage our source code.**

Course Structures

- Lectures
- Assignments
- Final Project
- Midterm + Final Exam

Grading Policy

Assignments $(6 * 5 \text{ points}) = 30$ → **30%**

Final Project 40 → **40%**

Midterm 10 → **10%**

Final Exam 20 → **20%**

Total 100 → **100 %**

Grading Policy

- Some assignments will be graded by giving a **live Demo** to your TAs.
- In each demo you need to **run** your code and **walk through your code** to explain it how you implemented key functions and components.
- You need to **submit your code by the due time** and then **sign up for the Demo grading**
- The due dates **will not be extended**.
- Late submissions will result in **a 20% deduction for the first 24 hours**. Submissions will **not be accepted after this timeframe**.

Course Information

- Instructor: Mahsa Saeidi
- Class time: Sundays and Tuesdays, 15-16:30
- Class location: 205
- Office hours: Sundays, 16:30-17:30
- Course Resources: <https://courses.aut.ac.ir>
- Communication:
 - Email: saeidi_it.86@aut.ac.ir
 - <https://courses.aut.ac.ir>
 - Office hours

Resources

- The majority of these slides have been adapted from materials provided by other instructors:
 - [lecture notes](#) developed by Mr. Parham Alvani
 - [lecture notes](#) developed by Dr. Rob Hess