IT1100 - Internet and Web Technologies

Web based Systems development

SLIIT
FACULTY OF COMPUTING

Content

- Distributed systems and their architectures
- Main concepts of web
- E-commerce systems and forms

Distributed systems and their architectures

Distributed systems and their Architectures

Computer based systems can be mainly divided into 2 types, *according to the distribution of the components*

- Standalone systems (or commonly referred as desktop applications)
- Distributed systems

Distributed Systems vs Standalone Systems

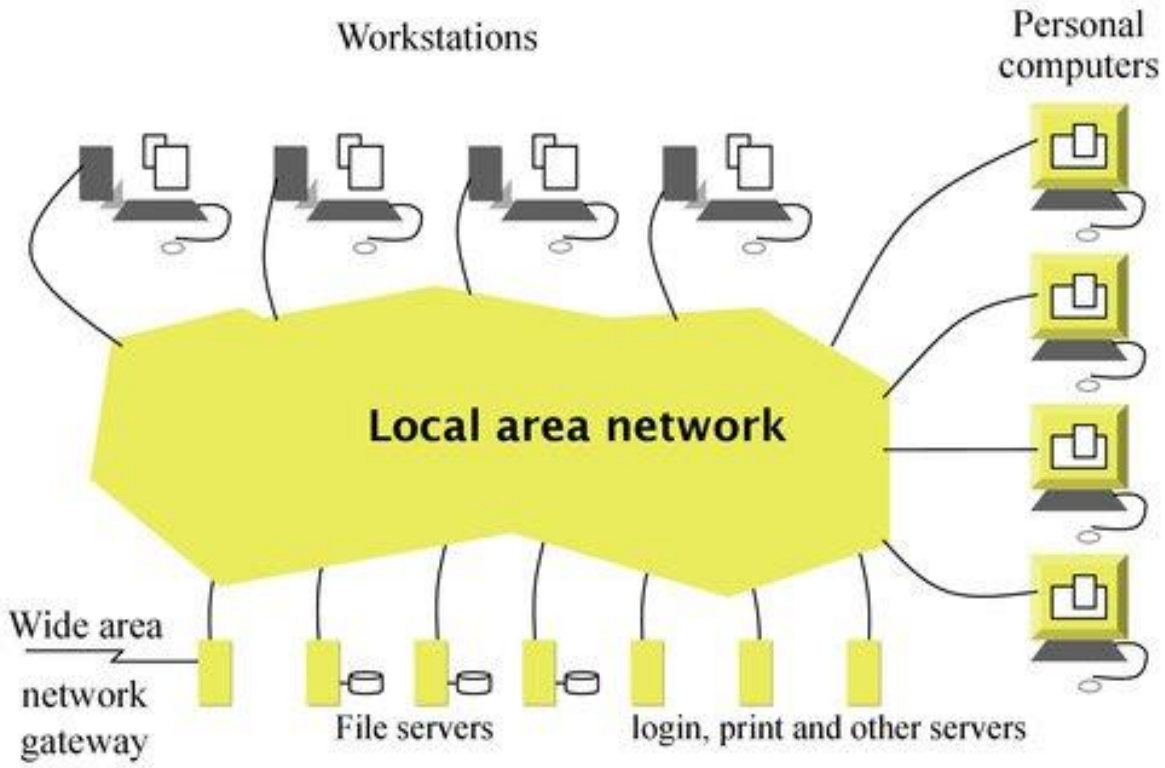
Standalone Computer System	Distributed system
▪ All the components are executed within a single device	▪ The components are distributed and executed in multiple devices
▪ Do not need a network	▪ Need a network
▪ Usually one or tightly coupled set of technologies are used to develop (JAVA, .NET)	▪ Multiple and loosely coupled set of technologies are used to develop (HTML+CSS+JS + PHP)

Distributed systems

- There are multiple, slightly different definitions and arguments about the terms **distributed systems** and **distributed computing**

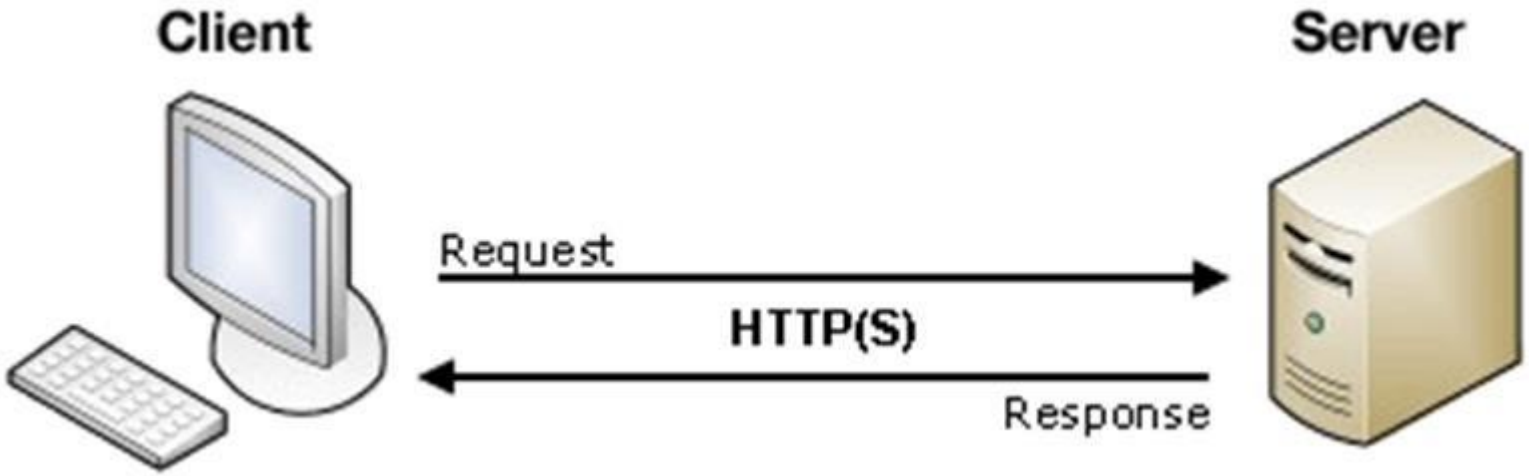
- We are here focusing on the systems, whose components are distributed among multiple devices and using a network for the communication between these components.

A Distributed System



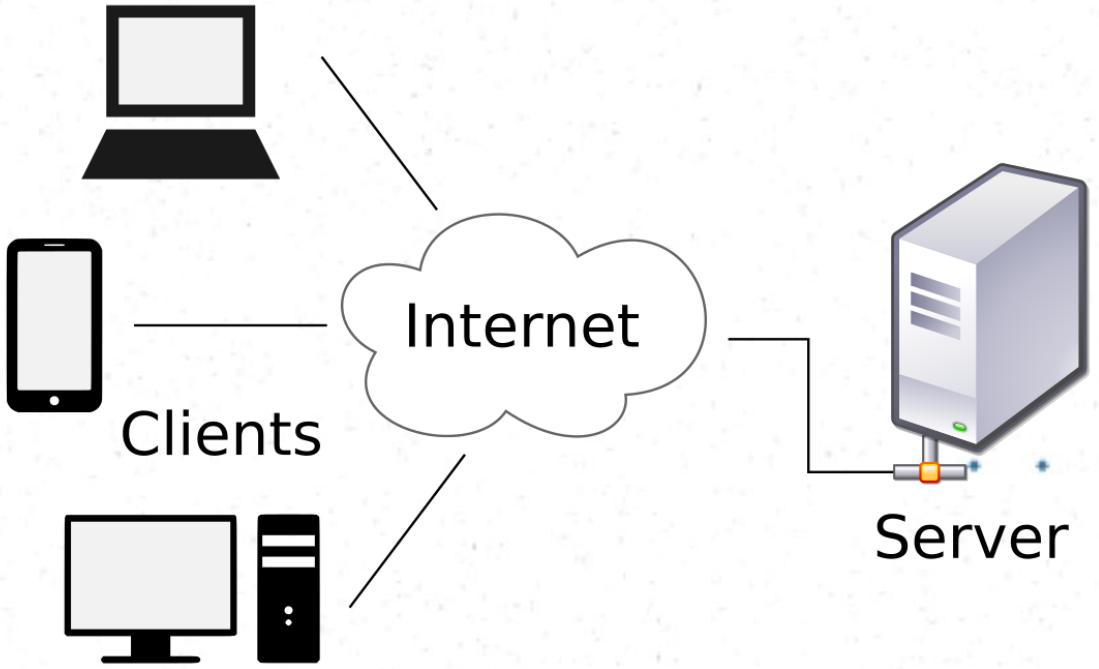
Client-server architecture (2-tier)

- The basic architecture of the distributed systems is called the client-server (or two-tier) architecture
- Usually the client (user) sends a **request** asking the server for some service and the server **responses** with the resources



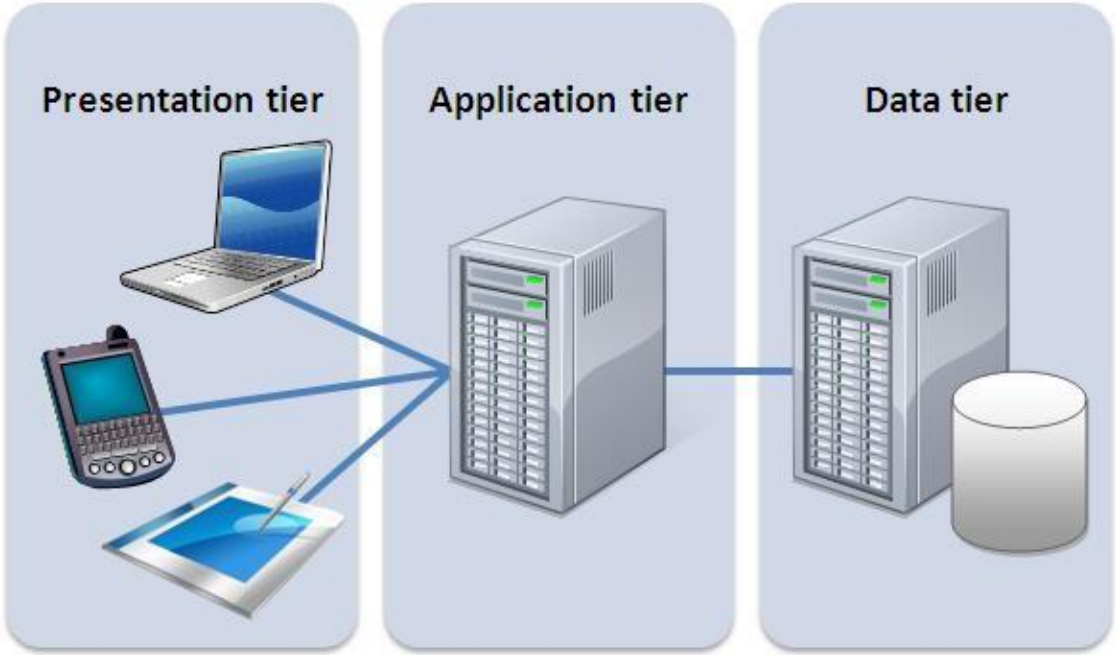
Client-server architecture (2-tier)

- There can be multiple clients, accessing the same server.
- These clients may use different types of devices



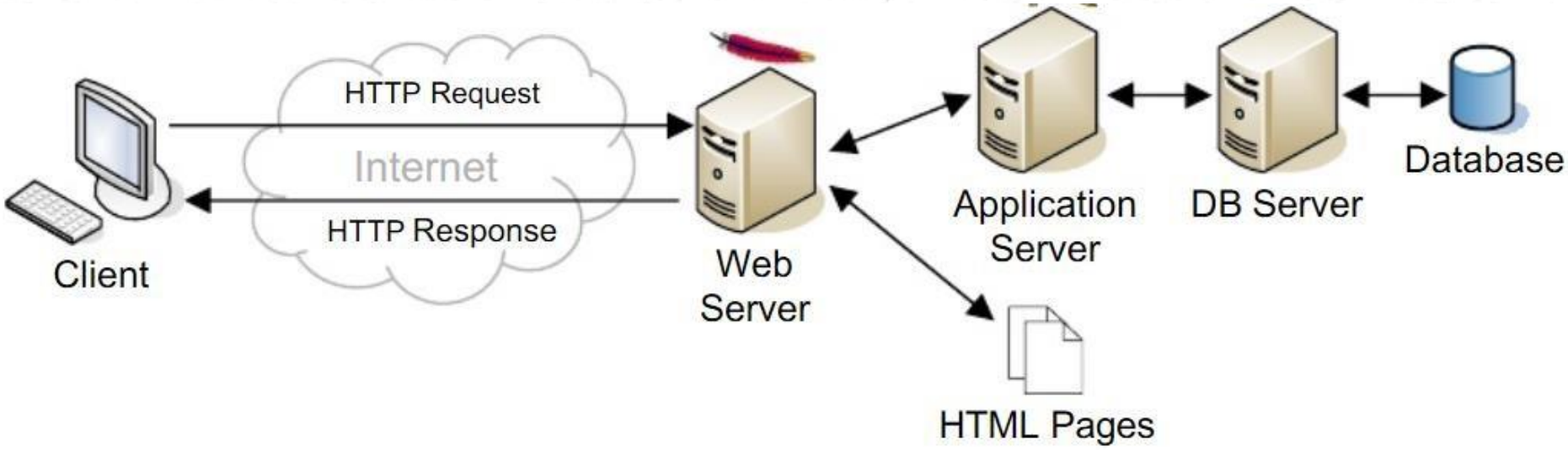
Client-server architecture (3-tier)

- 3-tier architecture is used, when there is a need for data persistence and also to separate the application logic from the data
- This can be seen as an extension of 2-tier architecture



Client-server architecture (n-tier)

- When there is a need for further separation and distribution of the components, more tiers can be added and extend the 2-tier or 3-tier architecture into an n-tier architecture



Main concepts of Web

Web server

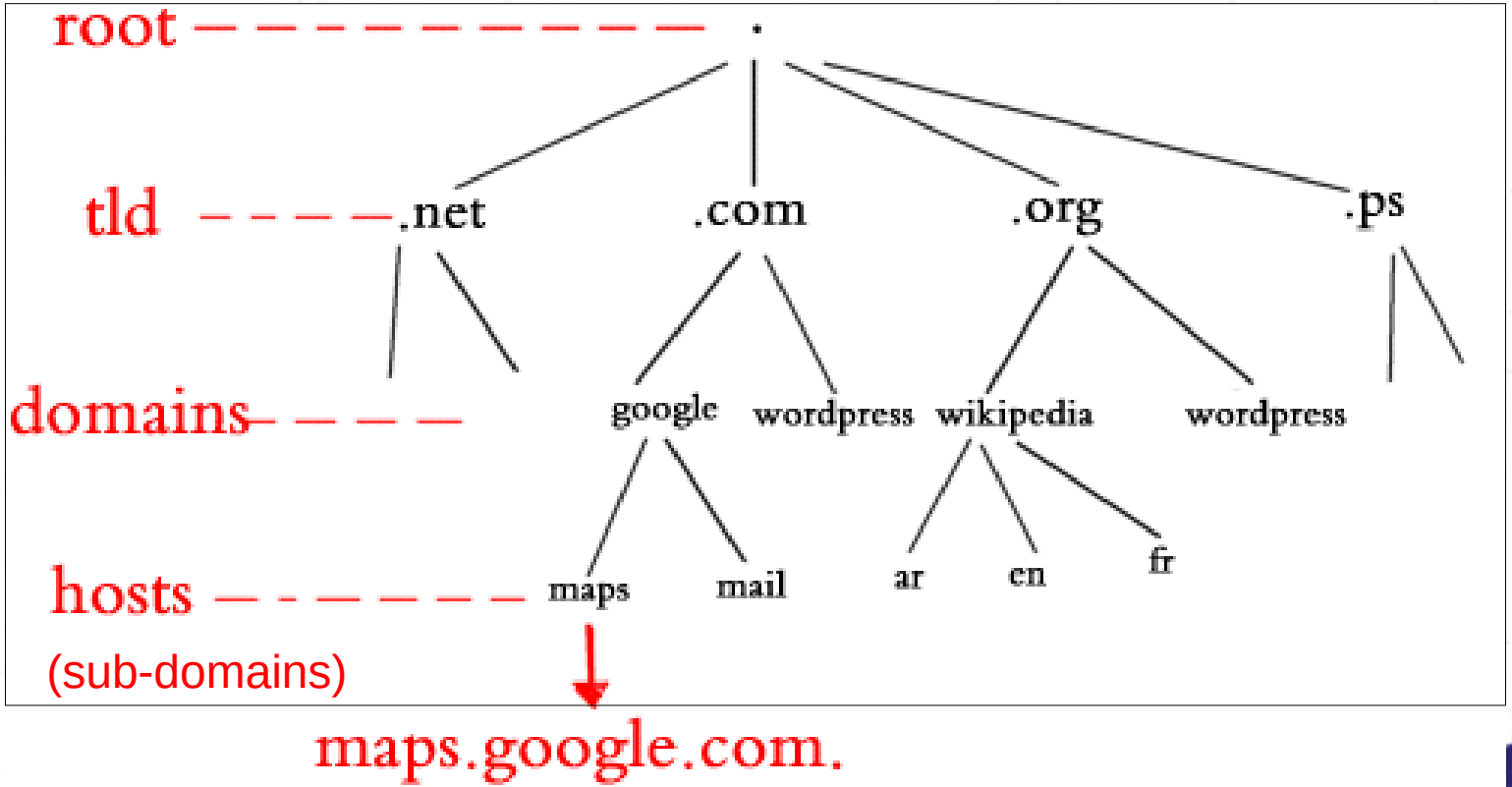
- Web server is a server software, which responses to the HTTP requests.
- Web server means
  - The server software
  - the hardware
  - other software infrastructures
    - which provide a platform to the server software to work and perform well.

Domain name

- The server computer has an IP address, which is used to access and communicate with the server.
  - Ex: 74.125.236.199
- An IP address is *not human friendly,* therefore more human friendly name is given for humans to identify the server, which is called the **domain name**
  - ex: www.google.com

Domain name

There is a hierarchy for the domain names

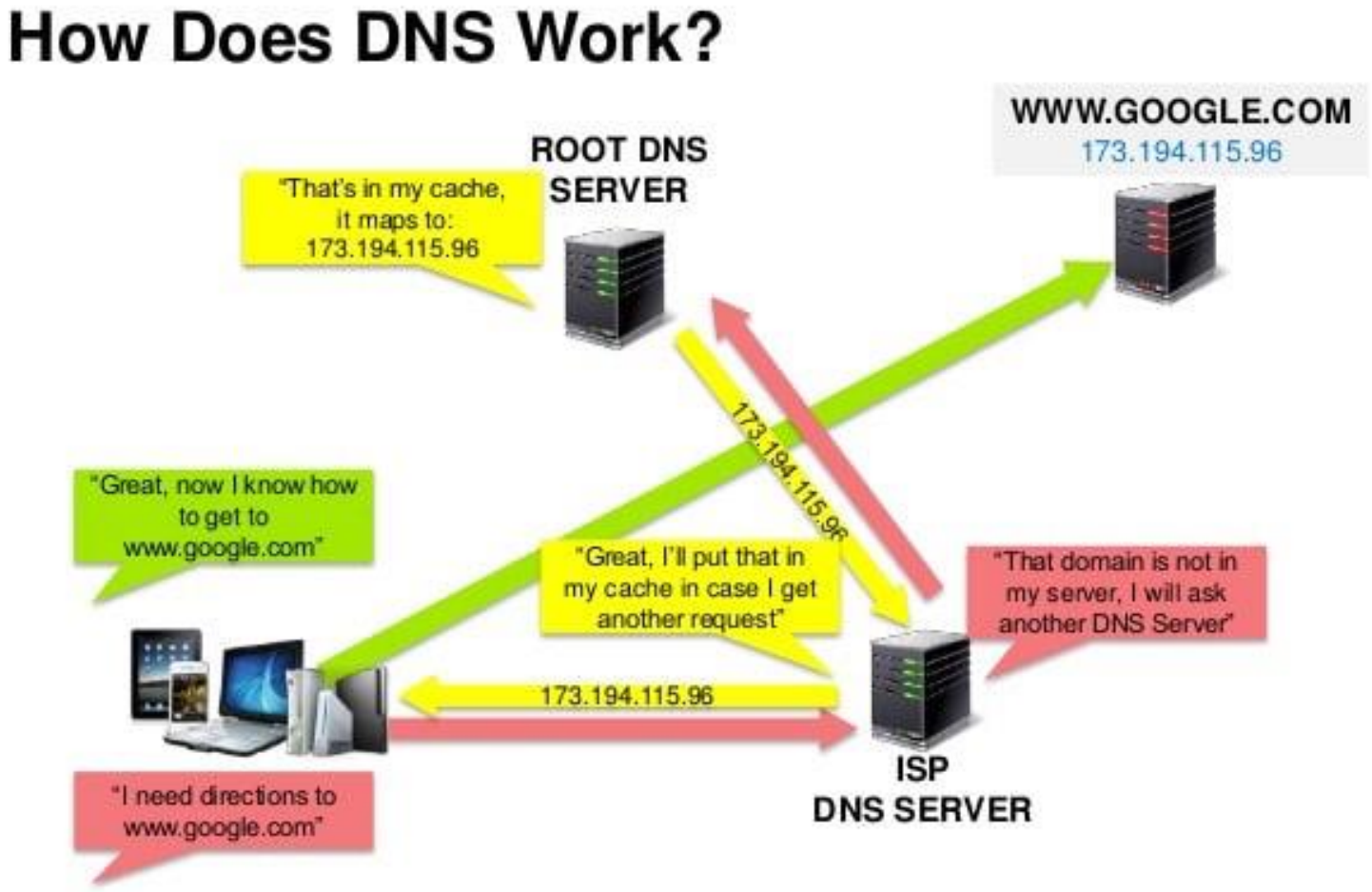


Domain Name System (DNS)

- DNS is a network, which consists of Domain Name Servers
- DNS helps to map the domain name to the IP address
  - This process is called the address resolution (DNS resolution)

https://en.wikipedia.org/wiki/Domain_Name_System

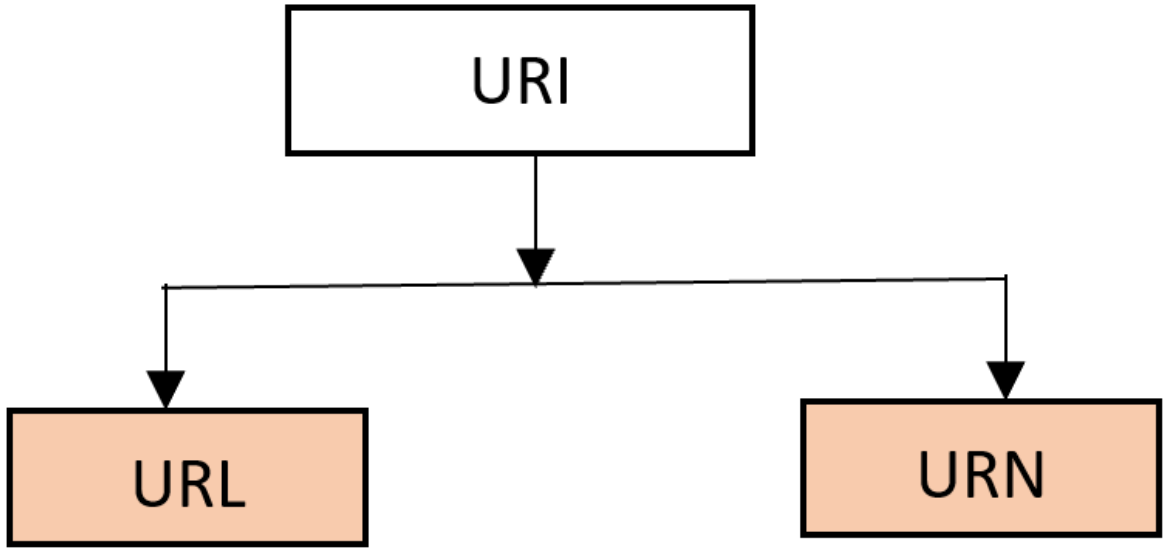
Domain Name System (DNS)



Unified Resource Identifier (URI)

- URI is a string of characters designed for **unambiguous identification** of resources.

- URI is extensible via the URI scheme



Unified Resource Identifier (URN)

- Unified Resource Name(URN) is a persistent, location-independent identifier

urn:nbn:de:101:3-2019075675872913

urn:uuid:6r4bc420-9c3a-12i9-97d9-0665700c9a66

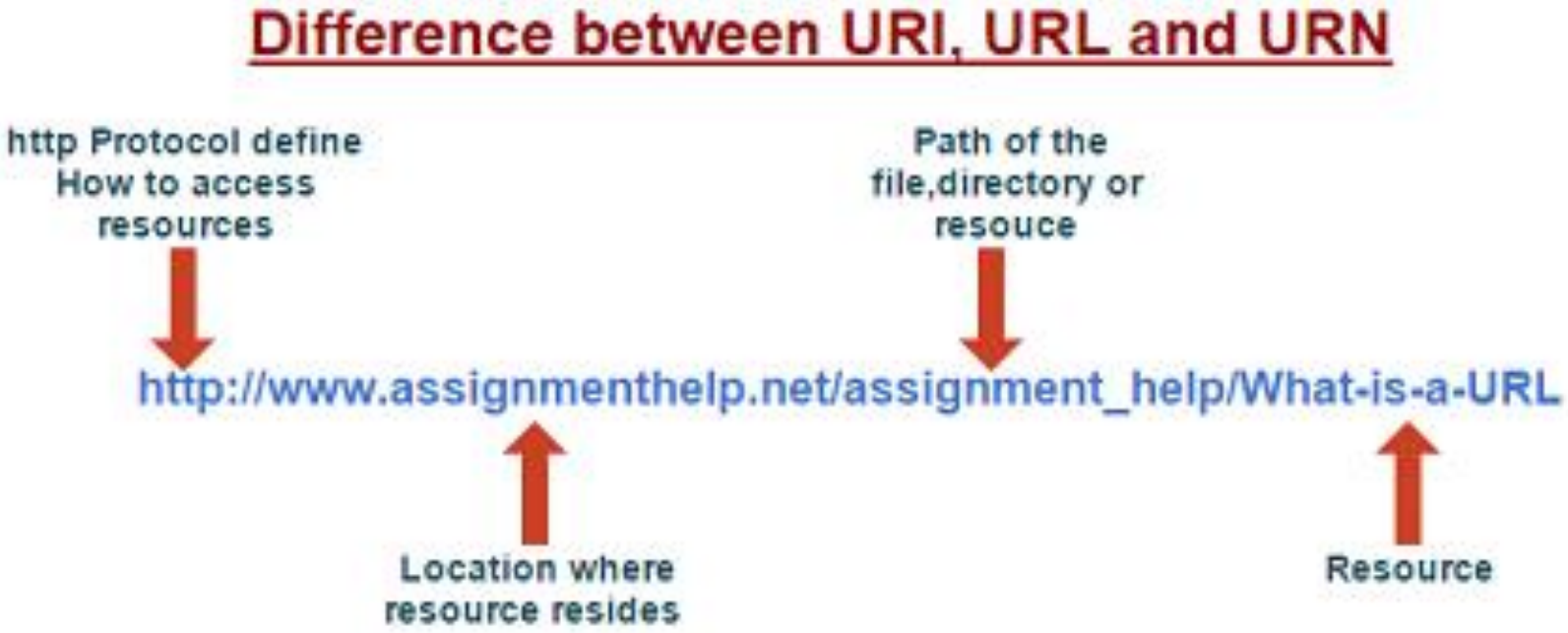
ISBN 1-446-2776877-40

Unified Resource Identifier (URL)

Unified Resource Locator (URL) can be seen as a web address, which is a reference to a web resource that specifies its location on a computer network.

- www.google.com/search?q=examples
- https://www.w3schools.com/html/default.asp
- https://en.wikipedia.org/wiki/URL

URI, URL and URN example



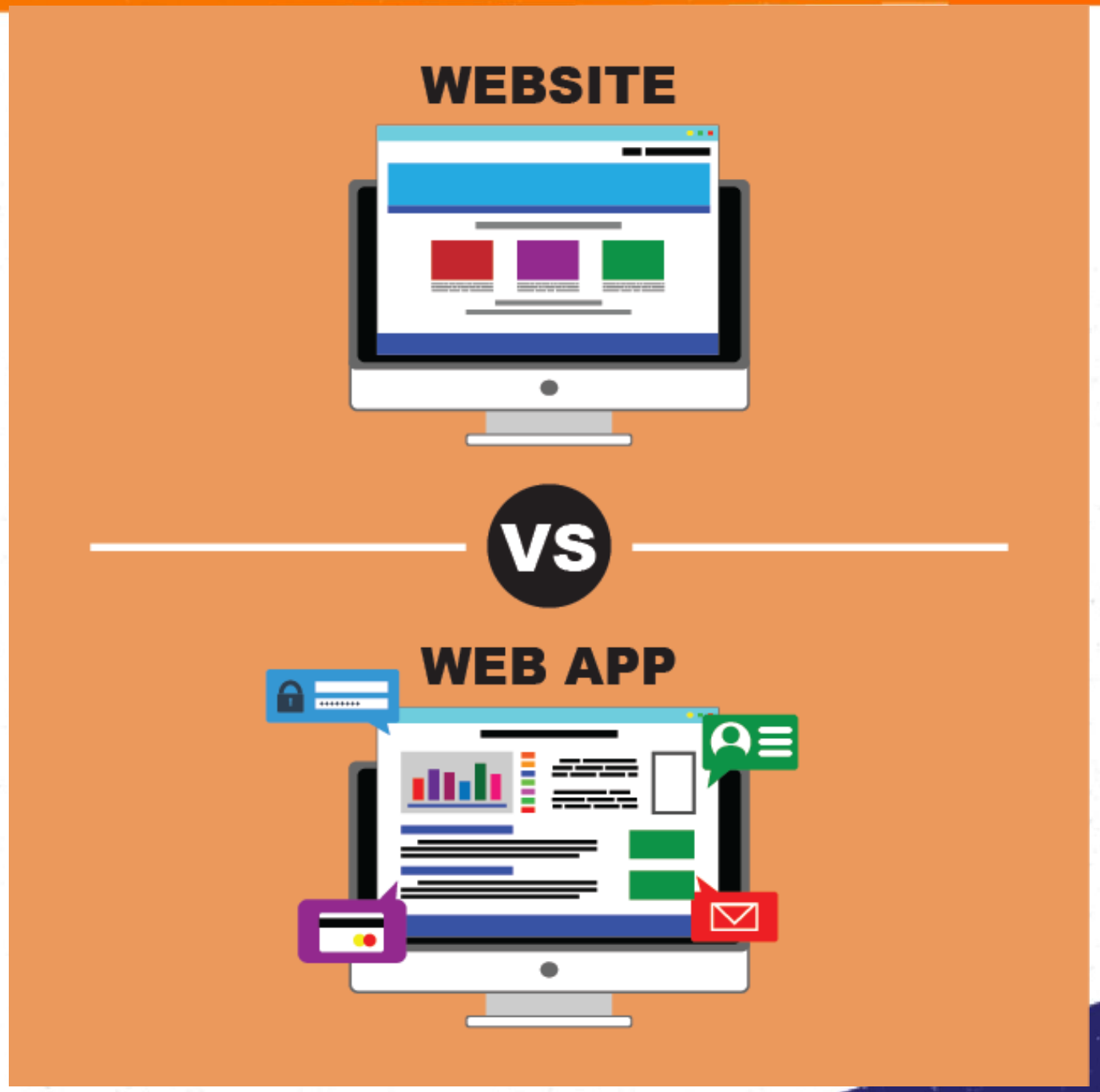
Websites

- **Website** can be seen as a collection of web pages with static content
- Early websites were entirely developed only using HTML
  - Nowadays, some server-side application components and databases are used to dynamically generate the content
  - However, still the content is not user tailored

Web Applications

- **Web application** is a single page or a collection of web pages, with *interactive components* to *dynamically* generate the content

- Users can enter data., process them, and get information as the result using a web application


WEBSITE
VS
WEB APP

Websites Vs. Web Applications

- The processing is done by the application components in
  - Client-side (in browser) [JS is used to develop]
  - Server-side (in web server) [PHP, Java can be used]
  - Both the client and the server sides

- These application components may allow the user to interact with the web application by entering data, selecting content, clicking, dragging and dropping, etc…

E-commerce systems and forms

E-commerce systems and forms

Nowadays, the web technologies are used to develop many types of applications, including

- Email (Gmail, yahoo mail, outlook)
- Office tools (Google docs, google sheets, etc..)
- Games (Poki, Miniclips)
- Multimedia (YouTube)
- Social media and networking (Facebook, Tweeter)
- E-commerce (Amazon, e-bay)
- And many more……..

E-commerce systems

E-commerce is a large domain, which covers many related concepts like

- Internet marketing
- Electronic fund transfer
- Online transaction processing
- And many more…..

E-commerce systems

E-commerce systems provide online buying and selling over the internet.

There is a large variety of types of e-commerce systems

- Online goods/soft items(software, e-books, videos)
- Retail services (travel, food, cloths)
- Marketing services (advertising, auctions)
- Customer services (help centers, online banking)
- Many more…..

E-commerce systems

- E-commerce systems can also be categorized according to the stakeholder engagement

– Business-to-Business (B2B)	– between companies
– Business-to-Consumer (B2C)	– traditional operations
– Consumer-to-Consumer (C2C)	– via an online platform

Advantages of e-commerce

- To customers
  - No queues
  - Reduced price
  - Global transactions
  - Available 24/7
  - Wide collection for easy selection

Disadvantages of e-commerce

- To customers
  - Cannot examine the product
  - Lacks the shopping experience
  - Needs internet access
  - Can be addicted

Advantages of e-commerce

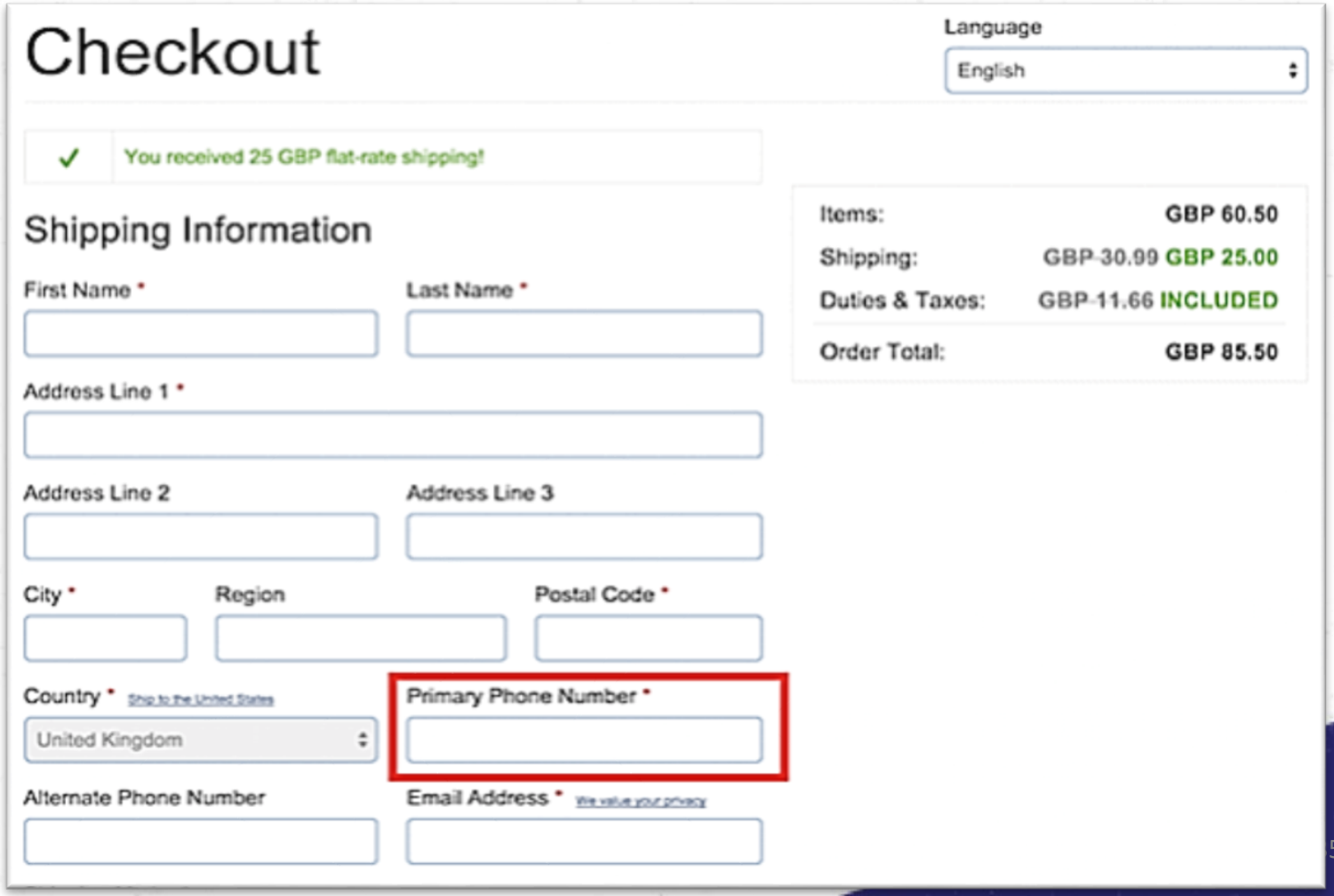
- *To businesses*
  - After the capital cost, maintenance cost is low
  - Global customers
  - Increased market share

Disadvantages of e-commerce

- *To businesses*
  - For physical items, storing and distributing is needed
  - Need to update the system frequently
  - Depends on the power and the internet

Web Forms

- E-commerce systems use web forms to capture the customers' data and send to the server for processing



Web Forms

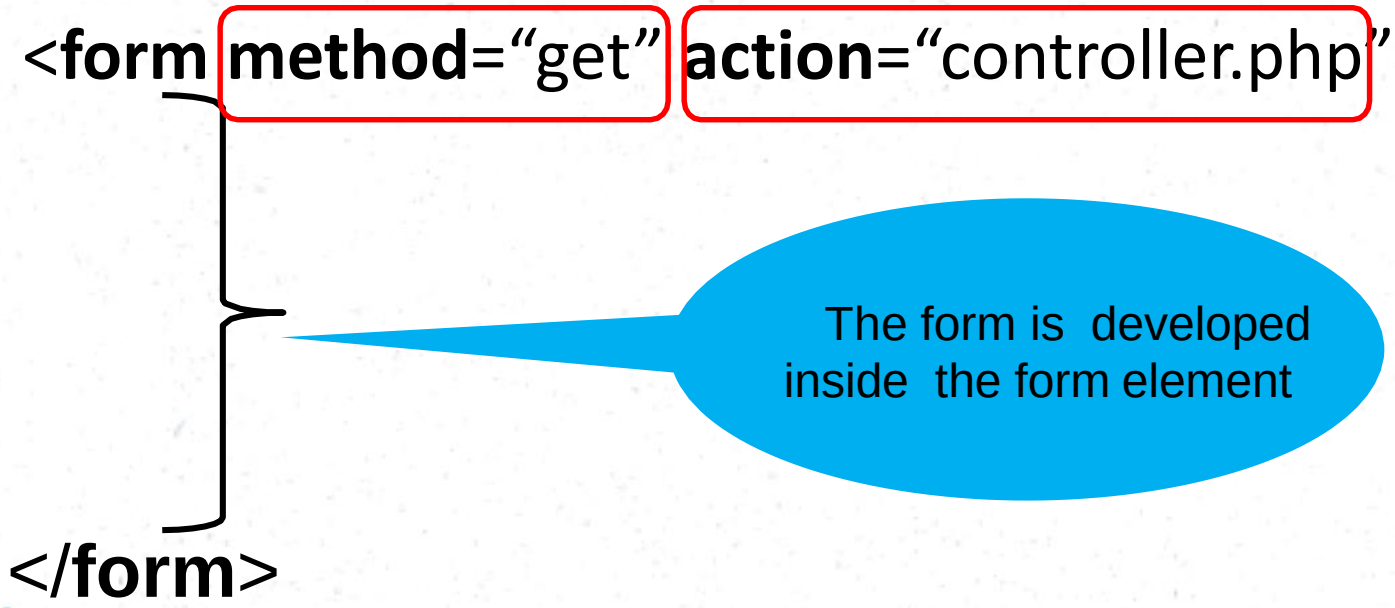
- Forms use variety of fields (elements) or structures not only to capture the users' data, but also to display data and information

- Input fields
  - Text boxes, dropdown lists, option buttons, radio buttons, selectable items, drag and drop items, file selectors, etc..

- Data/information display structures
  - Lists, tables, charts, images, files, etc..

Web Forms Development

- When developing HTML web forms, all the form elements are wrapped by the parent element named “form”, which has 2 main attributes



Form method

- Used to specify the type of the intended action the form submission is needed

Ex: available form methods in HTML

- When submitting data, the form method specifies the way the data should be submitted
  1. GET
  2. POST

Form method – get

- Default form method to submit data
- Data is visible on the address bar
- www.myDomain.com/controller.php?name=Saman&age=35
- Appends form-data into the URL in name/value pairs (Query string)
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
  - – GET is better for non-secure data
- Useful for form submissions where a user want to bookmark the result

Form method – post

- Use POST method to send sensitive or personal information.
  - The POST method does not display the submitted form data in the page address field.
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Form action

- Specifies the address (URL) to submit the form
  - Usually, a script file or a program
- This address points to a component in the server
- When the form is filled and submitted, this component will receive the form values then process and responds with the necessary output

Form submission

- There should be a submit button to submit the form to the action end, using the specified form method

```
<input type="submit" value="Submit">
```

Form elements

- Many form fields are developed using the input element

<input type="text">	Defines a one-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

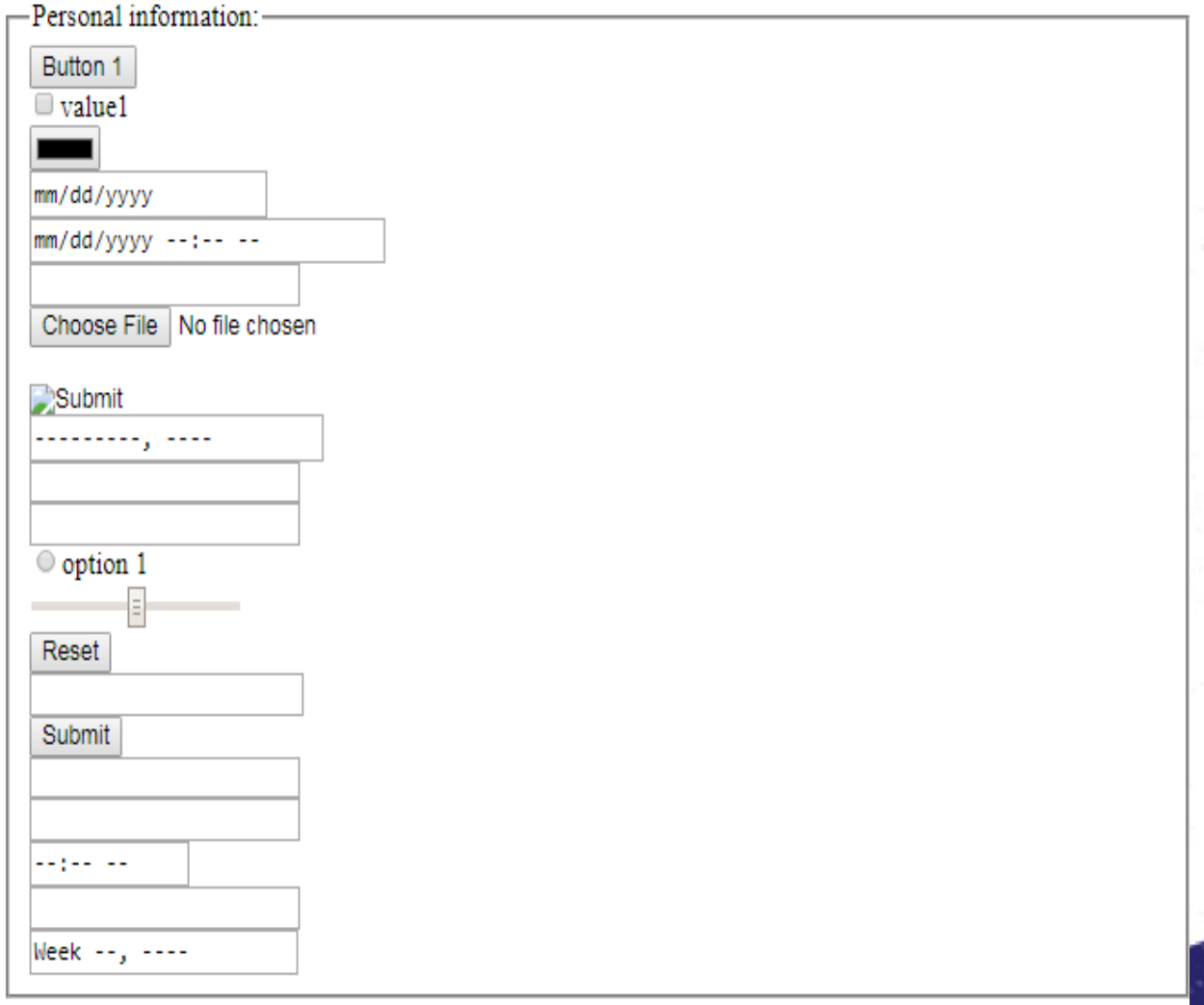
- You will learn more possible form elements and their use in the practical session

https://www.w3schools.com/htmL/html_forms.asp

```
<fieldset>
<legend>Personal information:</legend>

<input type="button" value="Button 1"><BR /
<input type="checkbox">value1<BR />
<input type="color"><BR />
<input type="date"><BR />
<input type="datetime-local"><BR />
<input type="email"><BR />
<input type="file"><BR />
<input type="hidden"><BR />
<input type="image"><BR />
<input type="month"><BR />
<input type="number"><BR />
<input type="password"><BR />
<input type="radio">option 1<BR />
<input type="range"><BR />
<input type="reset"><BR />
<input type="search"><BR />
<input type="submit"><BR />
<input type="tel"><BR />
<input type="text"><BR />
<input type="time"><BR />
<input type="url"><BR />
<input type="week"><BR />

</legend>
</fieldset>
```



Form validation

- It is very important to validate the data values entered into a form, before processing them

- Form validation can be done
  - Using HTML5 in **client-side** (before submitting the form)
  - Using JS in **client-side** (before submitting the form)
  - Using a **server-side** component (usually the component pointed by the form action) in the server (After submitting the form)

Form validation

Form validation – using HTML

<input type="text" name="name" **required**>

<input type="text" name="name" value="Sam" **readonly**>

<input type="text" name="name" value="Sam" **disabled** >

<input type="text" name="name" value="Sam" **size="20"** >

<input type="text" name="name" value="Sam" **maxlength="50"** >

- You will learn the use of JS and PHP to validate form data, when you learn JS and PHP

Summary

- Distributed systems and their architectures
- Main concepts of web
- E-commerce systems and forms