Web Essential

CT313H – WEB TECHNOLOGY

Objective

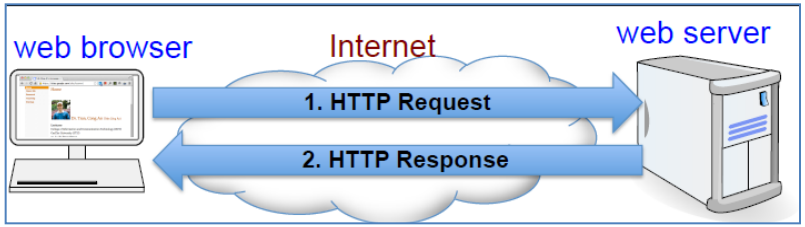
To provide students an overview about WWW
and essential knowledge
for web application development

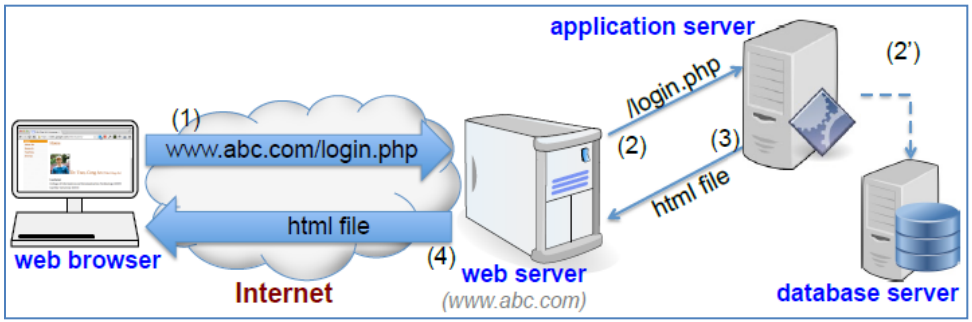
Content

- WWW and web applications introduction
- Basic concepts
- Client – Server model
- HTTP protocol
- Web technologies
- Web developer classification
- Pure Javascript web application

WWW and Web applications

- **World Wide Web (WWW)**:
  - Communication via HTTP
  - Document representation using HTML
  - Service architecture: Client – Server (2-tier)
- **Web applications** (dynamic web):
  - Applications that're built on WWW service
  - Server: performs calculations and returns the result in form of web pages ⇒ dynamic (web) content





Basic concepts

- Web browser/Web server
- Web hosting
- Web services
- Webpage, website, homepage
- TCP/IP
- HTTP
- DNS
- URI/URL
- HTML, CSS, JavaScript
- Ajax
- DOM
- XML
- JSON (JS Object Notation)
- REST/RESTful
- W3C, IETF, ICANN

Client – Server model

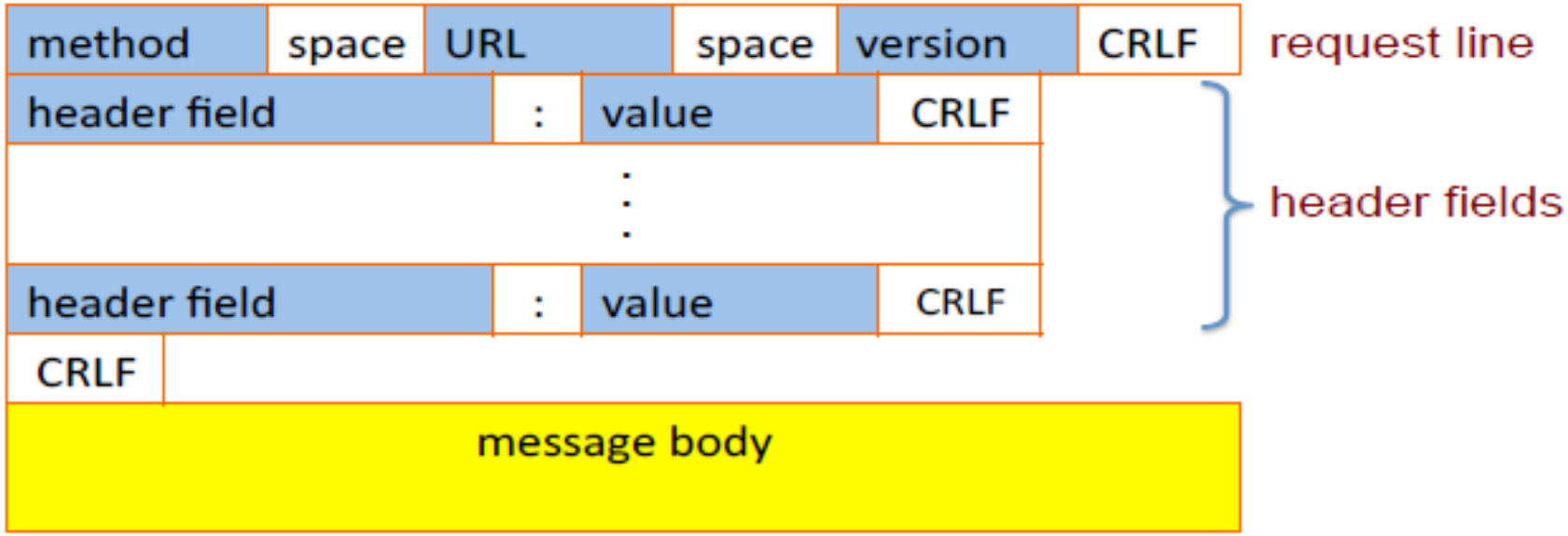
- **Server**: provides services
  - Listen requests from clients (on a particular port)
  - Processes and responses client's requests
  - Some web servers: Apache, IIS,… (default port: 80)
- **Client**: requests/consumes services
  - Provide UI to interact with user and get the user requests
  - Send user requests to server
  - Get response from the server and display the result to user
  - Some web clients (browsers): Chrome, IE, Firefox, Opera,…
- **Protocol**: a set of communication rules between Client and Server

HTTP protocol

- HTTP: **H**yper**T**ext **T**ransfer **P**rotocol
- Communication protocol of WWW
- A set of commands and rules used for communication between web browsers and web servers
- Data transmitted between web browser and web server is often pure text, particular hypertext documents
- This is a stateless protocol: server is not required to remember anything about client between requests
- HTTP versions: 0.9, 1.0, and 1.1 (lastest)

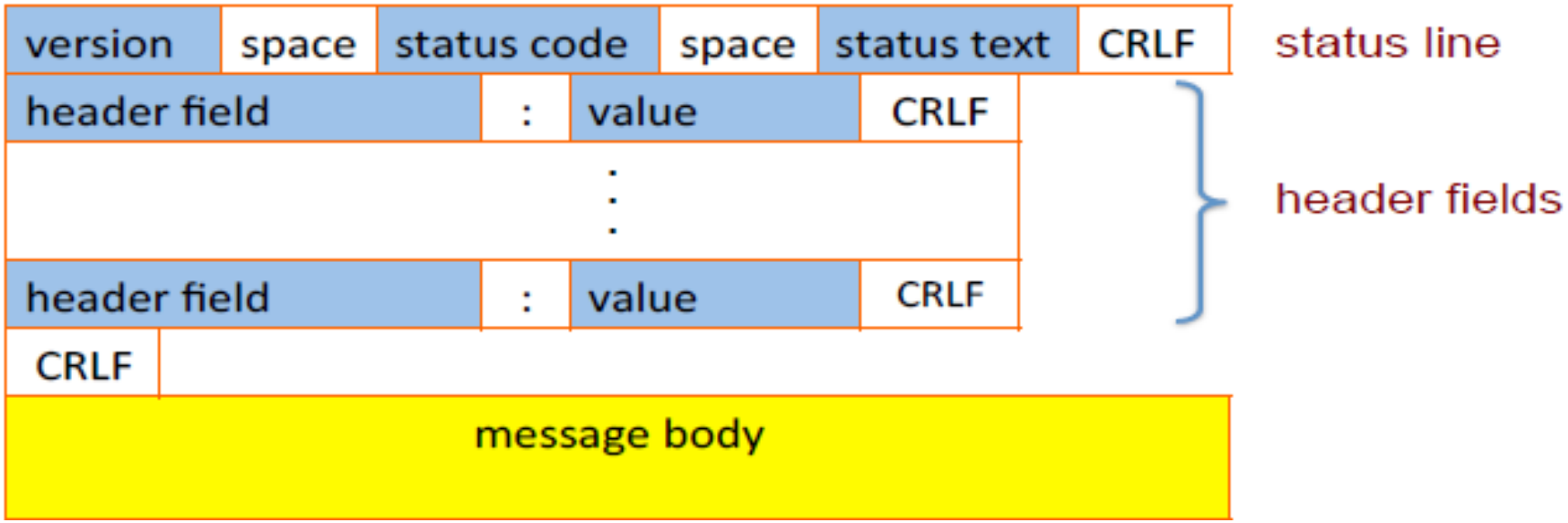
Structure of a request

- **Methods**: GET, POST, PUT, DELETE, OPTIONS, HEAD
- **Header fields**: Accept, Content-Length, Content-Encodeing, Accept-Language,…

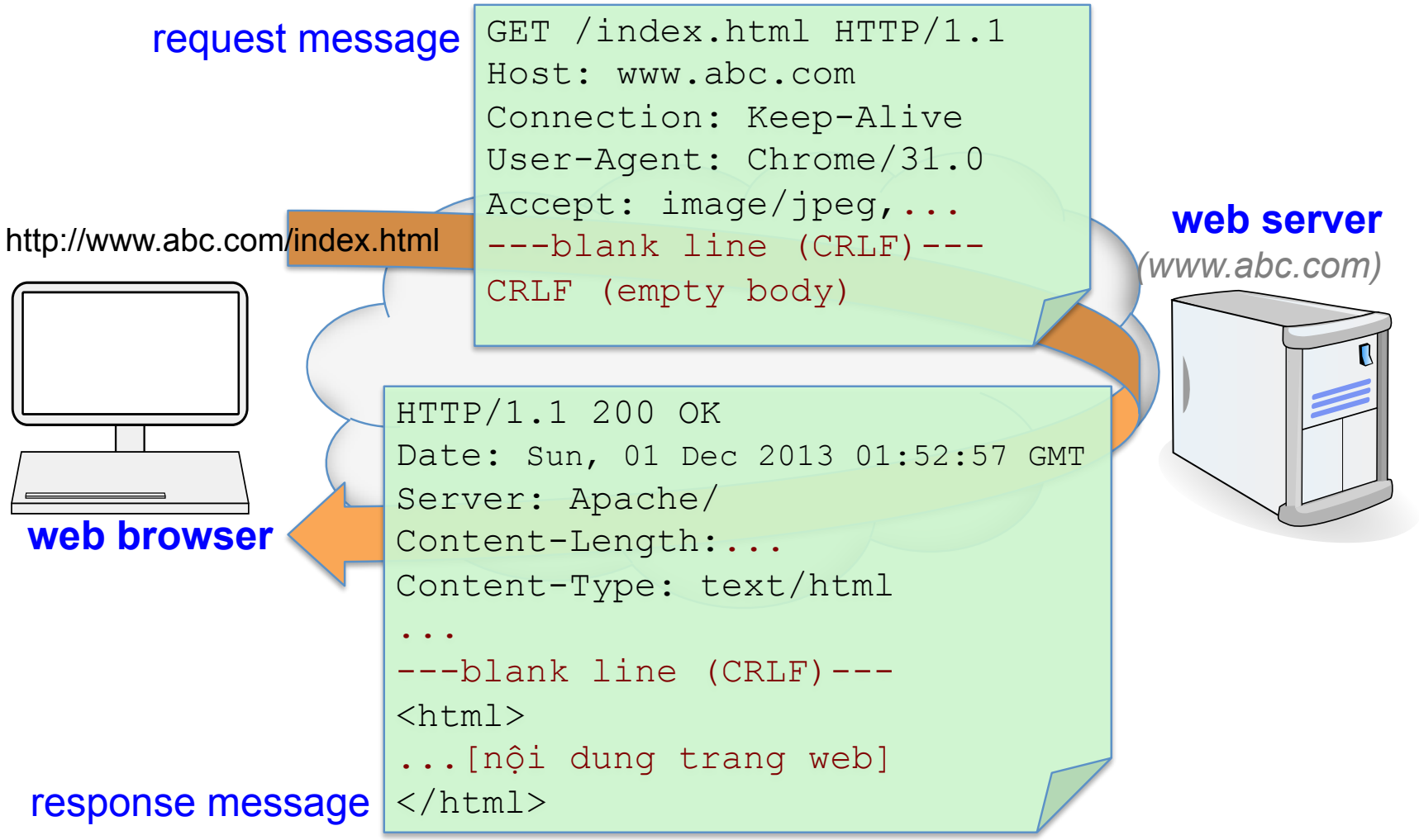


Structure of a response

- **Status codes**: 200 (OK), 301 (moved permanently), 401 (unauthorized), 404 (not found), 500 (internal server error)
- **Header fields**: similar to the request message



Example


request message
GET /index.html HTTP/1.1
Host: www.abc.com
Connection: Keep-Alive
User-Agent: Chrome/31.0
Accept: image/jpeg,...
---blank line (CRLF)---
CRLF (empty body)
http://www.abc.com/index.html
web server
(www.abc.com)
web browser
HTTP/1.1 200 OK
Date: Sun, 01 Dec 2013 01:52:57 GMT
Server: Apache/
Content-Length:...
Content-Type: text/html
...
---blank line (CRLF)---
<html>
...[nội dung trang web]
</html>
response message

Modern Web technologies

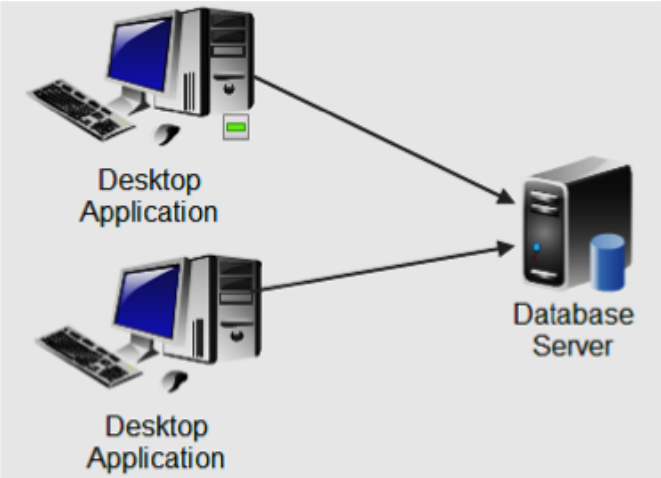
- Client side (Front-end):
  - HTML, CSS, JavaScript, AJAX,…
  - Bootstrap, jQuery, AngularJS,…
- Server side (Back-end):
  - PHP, JSP, Python, Ruby on Rails, ASP.NET, NodeJS,…
- Web development tools:
  - Bower: package manager
  - Grunt: JavaScript Task Runner, provides automation for NodeJS projects (e.g. minification, compilation, unit testing)
  - Yeoman: the web's scaffolding tool for modern webapps, used to create structure for a new project

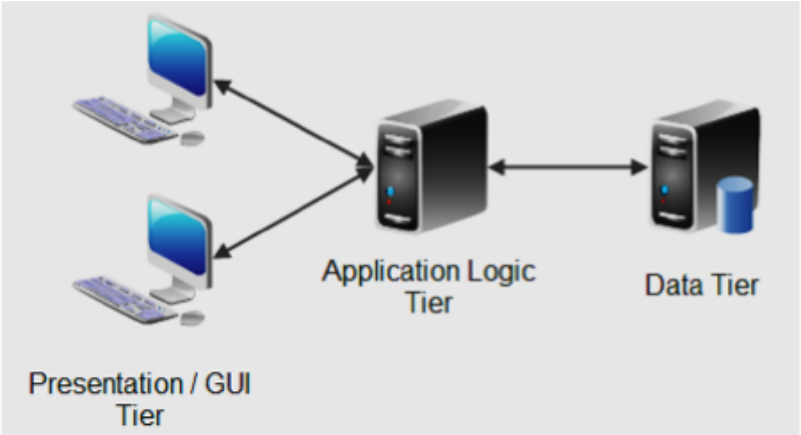
Web developer classification

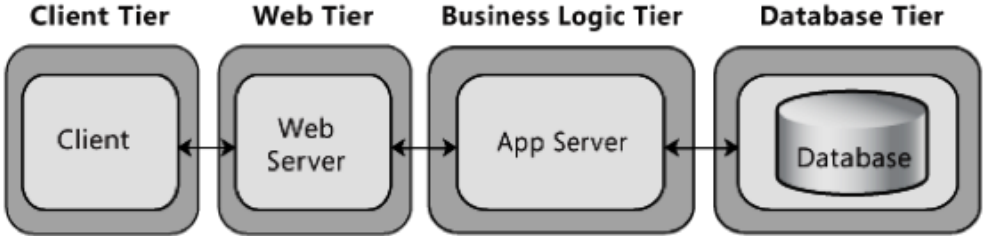
- Front-end developers:
  - UI design, communicate with users at browser
  - Technologies: HTML, JavaScript, image processing, CSS,…
- Back-end developers:
  - Process businesss logic at server
  - Technoligies: HTML, PHP/ASP/Java/JavaScript/Python/Ruby-on-Rails/…, SQL, web tools,…
- Full-stack developers:
  - Combination of front-end và back-end

n-tier architecture

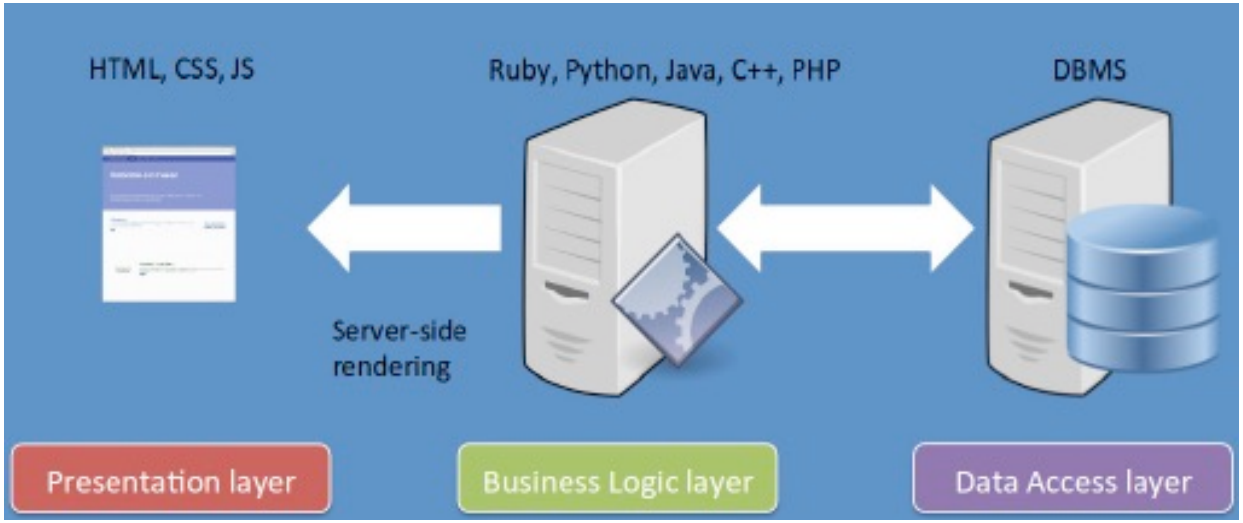
- Traditional client-server: 2 tiers
- Modern web applications: from 3 tiers or more (n-tier architecture)

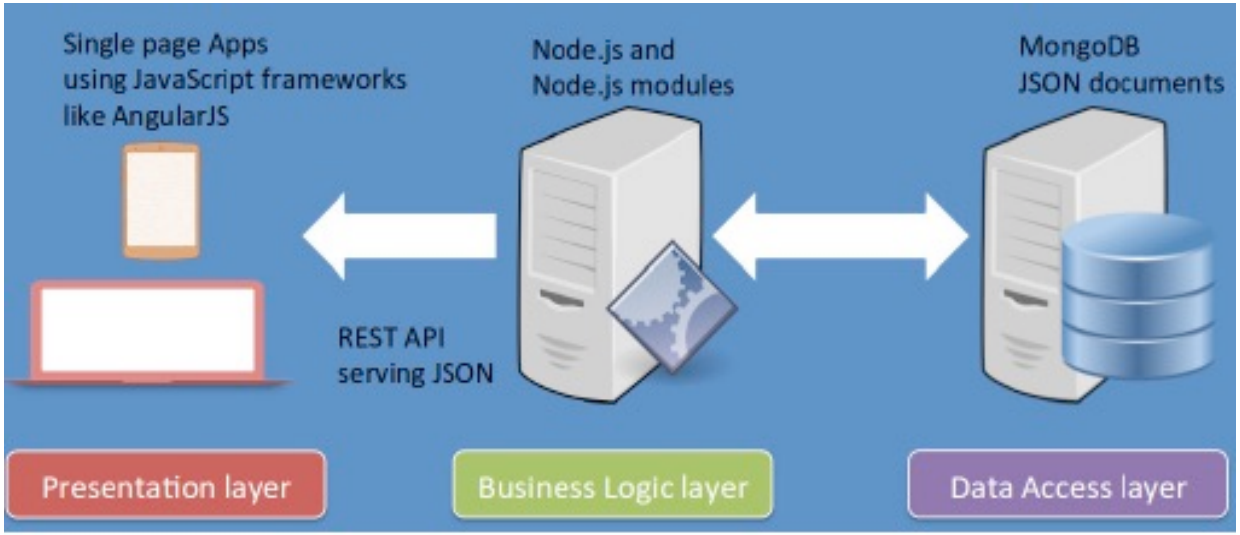






Pure Javascript web applications





Pure Javascript web applications

- Advantages:
  - Easy share code between client and server
  - Asynchronous event driven IO helps concurrent request handling.
  - npm (Node Package Manager): one of the biggest package managers
  - Possible to stream large files
  - JSON supported
- Disadvantages:
  - Not suited for CPU-intensive tasks (web server: I/O-intensive)
  - Lack a standardization (*)

Question?

CT313H – WEB TECHNOLOGY