TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING

WEB APPLICATION DEVELOPMENT USING NODEJS - 502070

LAB SESSION 1

By Mai Van Manh

OBJECTIVES

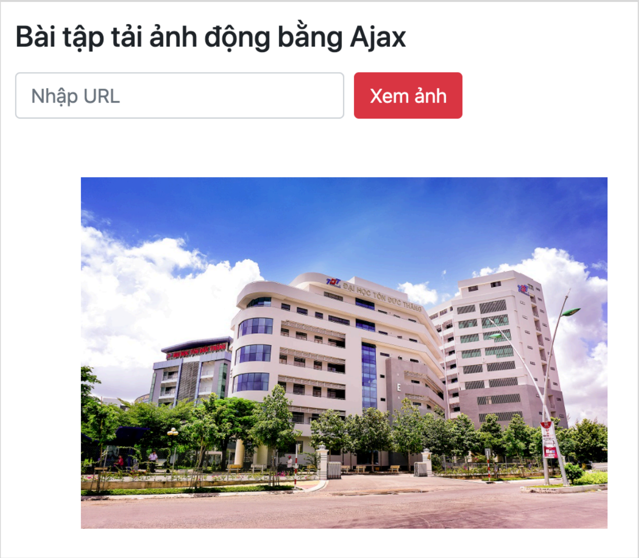
1. Review fundamental HTML and Javascript concepts studied in previous courses.
2. Revisit knowledge related to Ajax and the Document Object Model.
3. Explore several new features introduced in the JavaScript 2015 version, such as fetch, promises, async, and await.
4. Investigate some APIs introduced alongside HTML5, including local storage and session storage.

The following exercises aim to reinforce essential knowledge of HTML5/Javascript and can be directly performed within an HTML file, without the need for NodeJS.

Exercise 1. Ajax (XMLHttpRequest) and Document Object Model (DOM) Exercise.

Complete the following tasks:

1. Input the URL of an image file.
2. Upon clicking the 'View Image' button, utilize Ajax to fetch the image file as a **blob**.
3. Create an **HTMLImageElement** object (img tag) and display the downloaded image on the webpage (use `URL.createObjectURL(blob)`).
4. Upon subsequent clicks of the "View Image" button, as the img element has been created, only replace the src attribute with the new blob value.
5. If the image is clicked, download the image immediately without any confirmation or dialog.



Bài tập tải ảnh động bằng Ajax

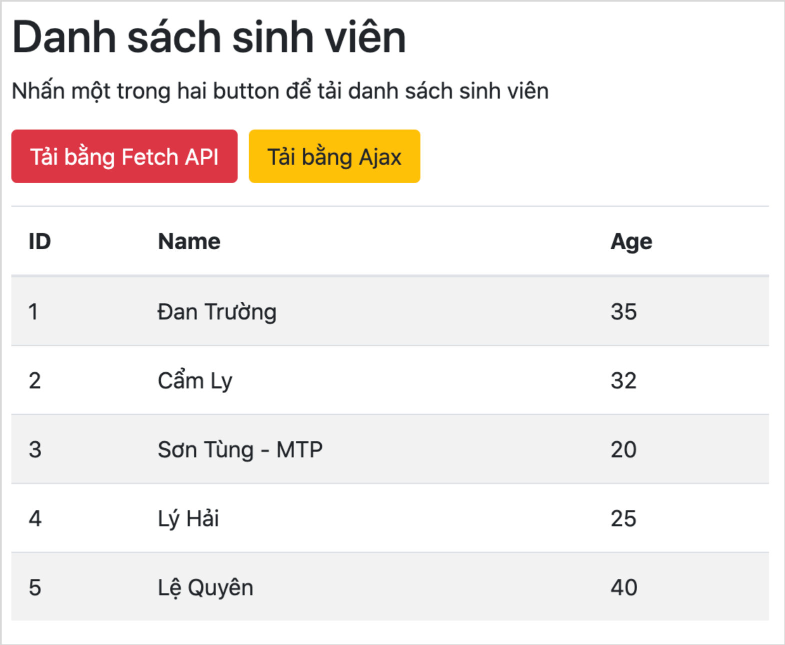
Use Ajax to load the image data; do not rely on the browser's default loading mechanism.

Note: In certain cases, you might encounter the "No 'Access-Control-Allow-Origin' header" error while loading images. Please use the following URLs:

- https://web-502070.web.app/lab1/tdtu1.jpg
- https://web-502070.web.app/lab1/tdtu2.jpg
- https://web-502070.web.app/lab1/tdtu3.jpg
- https://web-502070.web.app/lab1/tdtu4.jpg

Exercise 2. Fetch API and Ajax Exercise

Given data containing a list of students stored in a JSON file at the following link: https://web-502070.web.app/lab1/students.json. Utilize the Fetch API and Ajax to fetch the data and then display it in an HTML table.



Danh sách sinh viên

Nhấn một trong hai button để tải danh sách sinh viên

ID	Name	Age
1	Đan Trường	35
2	Cẩm Ly	32
3	Sơn Tùng - MTP	20
4	Lý Hải	25
5	Lệ Quyên	40

Expected outcome upon successful data retrieval

Exercise 3. Promise and Async, Await Exercise

The image loading functionality in Exercise 1 is currently implemented using Ajax. Adjust the source code to encapsulate this functionality within a Promise object.

Then, execute the created Promise in two ways:

- Conventional approach: Promise.then(...).catch(...)
- Utilizing async and await in conjunction with try/catch..

Exercise 4. Exercise on Local Storage and Session Storage

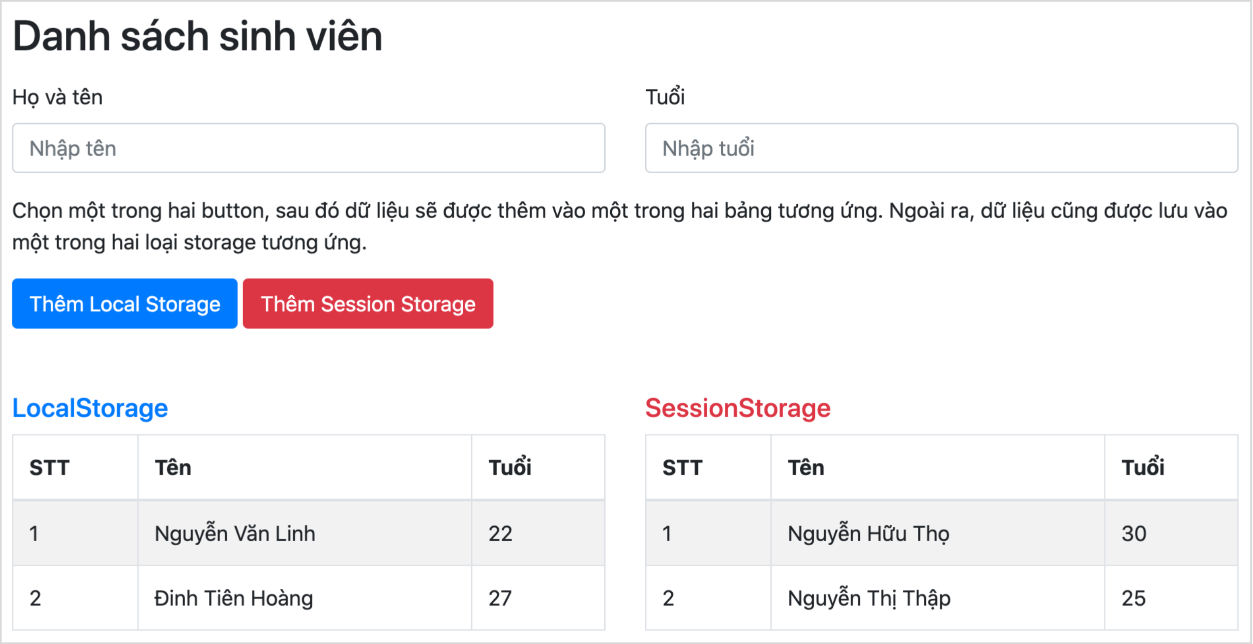
Local Storage and Session Storage are two data storage mechanisms provided in HTML5 and accessed through the Javascript API. Each browser may support different storage capacities, but typically Local Storage offers 5 MB while Session Storage provides 10 MB of storage.

- Local Storage: Shared across the entire website, meaning data stored on localhost/index.html can be read by localhost/home.html (and vice versa). Data stored in local storage persists even after closing the browser or turning off the computer. It does not expire automatically and requires manual deletion.
- Session Storage functions similarly to Local Storage, differing only in scope. Each browser tab is granted a separate session storage, which is not shared between tabs. Data in session storage remains after page reload (F5) and is lost only upon completely closing the tab.
- Cookie: Unlike local storage and session storage, cookies allow storing up to 4KB of data, and the cookie data is sent with every http request.

For Local Storage and Session Storage to function, the web page must be loaded using http or https; they won't work with the file:// protocol (meaning double-clicking the HTML file).

Create a web page for managing and storing student data in local storage and session storage.

- The web page allows users to input basic student information, which is then displayed in two corresponding tables.
- Moreover, the added data should be automatically stored in the respective storage types.
- When the page is reloaded or a new tab is opened, if data has been stored in the respective storage types, it should be loaded and displayed in the corresponding tables.



Danh sách sinh viên

Họ và tên: Nhập tên

Tuổi: Nhập tuổi

Chọn một trong hai button, sau đó dữ liệu sẽ được thêm vào một trong hai bảng tương ứng. Ngoài ra, dữ liệu cũng được lưu vào một trong hai loại storage tương ứng.

LocalStorage

STT	Tên	Tuổi
1	Nguyễn Văn Linh	22
2	Đinh Tiên Hoàng	27

SessionStorage

STT	Tên	Tuổi
1	Nguyễn Hữu Thọ	30
2	Nguyễn Thị Thập	25

Illustrative image of the requested web page design