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

WEB APPLICATION DEVELOPMENT USING NODEJS - 502070

LAB SESSION 6-7

By Mai Van Manh

OBJECTIVES

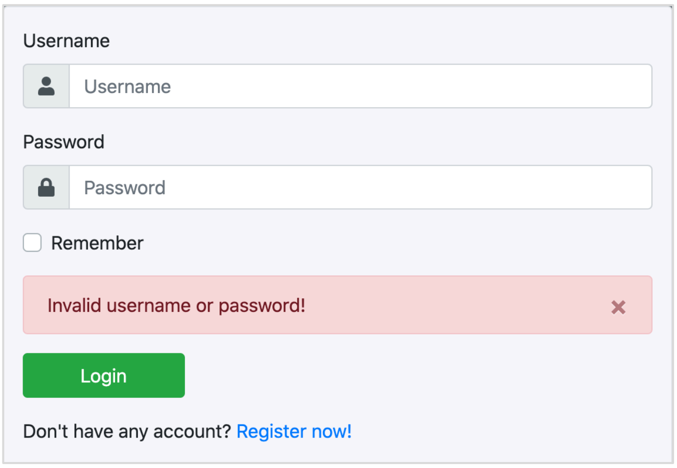
1. Implement login functionality using cookies and sessions in Express.
2. Employ Express static to serve static files on the server.
3. Utilize the fs module for reading and writing files on the server.
4. Explore mechanisms for directly delivering files via HTTP Response, utilizing headers like Content-Type and Content-Disposition. Learn how to throttle file download speed through HTTP.
5. Establish connections to and interact with MySQL databases using either mysql or promise-mysql.
6. Utilize bcrypt for hashing user passwords.
7. Utilize cross-site request forgery tokens to safeguard against CSRF attacks.
8. Combine ajax/fetch with server-side REST API to execute features that involve reading/changing data from the server without needing to reload the webpage.

ASSIGNMENT DESCRIPTION

Utilize ExpressJS and necessary modules to develop a file management web application. The web application should encompass features such as account registration, login, and file/folder management.

- Account Registration: Required information includes username, email address, and password. This data needs to be stored in a MySQL database. Passwords should be hashed using algorithms like bcrypt before storage. Both the registration/login forms need to employ a CSRF token to prevent cross-site request forgery attacks.
- Login: Use previously registered accounts for logging in. If a user who isn't logged in accesses the homepage, they should be automatically redirected to the login page. If

a user who is already logged in attempts to access the login/registration page, they should be automatically redirected to the homepage.



Username

Username

Password

Password

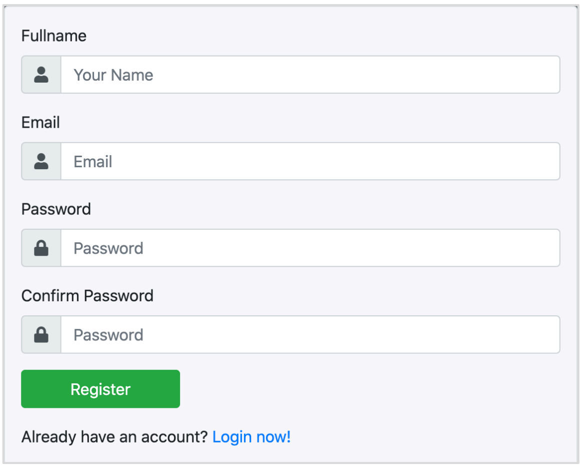
Remember

Invalid username or password!

Login

Don't have any account? Register now!

Login page interface



Fullname

Your Name

Email

Email

Password

Password

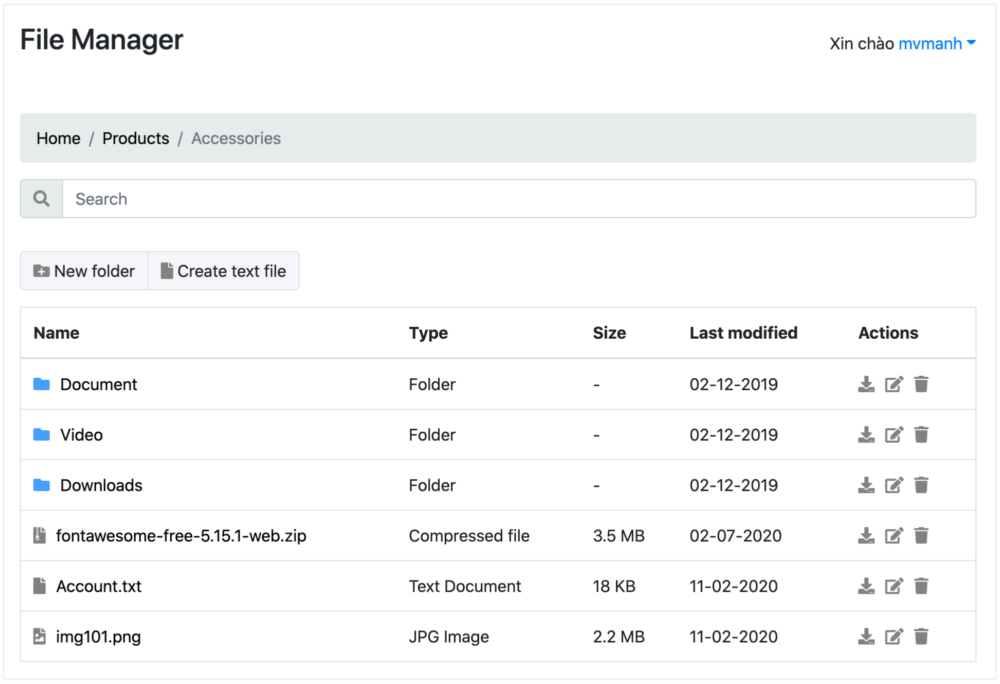
Confirm Password

Password

Register

Already have an account? Login now!

Registration page interface



File Manager

Xin chào mvmanh

Home / Products / Accessories

Search

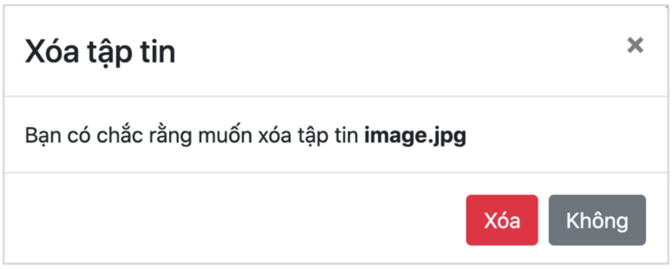
New folder | Create text file

Name	Type	Size	Last modified	Actions
Document	Folder	-	02-12-2019	
Video	Folder	-	02-12-2019	
Downloads	Folder	-	02-12-2019	
fontawesome-free-5.15.1-web.zip	Compressed file	3.5 MB	02-07-2020	
Account.txt	Text Document	18 KB	11-02-2020	
img101.png	JPG Image	2.2 MB	11-02-2020	

File/Folder Management interface on the homepage

- Homepage: The homepage displays an interface for users to manage files and folders. This page provides functionalities such as:

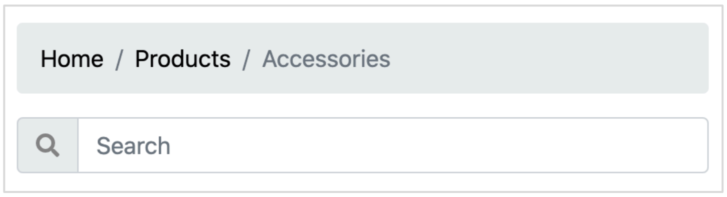
- Listing file and folder contents: Arrange folders first, followed by files. Display additional information like icons, file types, sizes, update dates, etc.
- Viewing files: Clicking on a file opens or downloads it (utilize Express static for this feature).
- Uploading a new file: Use ajax/fetch to upload files and display a progress bar during the upload process. Only one file needs to be uploaded at a time.
- Creating a new folder, renaming files/folders, deleting files/folders: Display appropriate confirm dialogs before deletion/data modification.


Xóa tập tin

Bạn có chắc rằng muốn xóa tập tin **image.jpg**

A confirmation dialog before deleting a file

- Searching for files and folders: As keywords are typed, results are filtered and only relevant matches are shown in the file list.
- Downloading files/folders: Clicking the download icon next to each file/folder triggers a download without leaving the page. If it's a folder, the web application downloads a zip file of the folder. After successful download, the zip file of the folder is automatically deleted on the server (utilize headers like Content-Type, Content-Disposition to implement this feature). Additionally, download speed limits can be implemented.
- Navigating between folders: Clicking on any folder reveals its subfiles and subfolders. Furthermore, when in any folder, users can quickly navigate to the root folder or any of the parent folders (via Bootstrap Breadcrumb).



The path bar and the search keyword input box

Other requirements

- From a source code perspective: Use Express Router to create separate routes for the two features: account management and file management (e.g., AccountRouter and FileRouter).
- Features in the file management interface (homepage) should be executed using ajax/fetch to avoid page reloading.
- If users access paths other than those described above, they should be redirected to the /error path and a 404 not found error interface should be displayed..
- Continue using modules from previous practical sessions and apply them to this exercise. For instance:
  - Use sessions and cookies to remember login information.
  - Use express-form, express-validator to validate data from HTML forms.
  - Use multer to handle file uploads.
  - Use flash messages to communicate notifications between routes.
  - Use the rate-limiting feature in Express to counter DDOS attacks.