MSC – CC – Assignment 03 - Dropbox

Brandon Rozario

Student No: 2955533

1. UI

- Design Philosophy: simple, minimal, and clean

- Unlike the previous 2 assignments that have loads of pages and redirects, this one has just one web page to keep things incredibly simple.

- A single HTML page contains everything that the application has to offer. This includes adding files, uploading folders, bookmarking, renaming, navigating, and deleting.

- Error messages pop up as browser alerts to get the user's attention.

- Icons are borrowed from Google's material design (https://material.io/icons/) to make the page aesthetically pleasing.
  This also eliminates the need of textual explanation since a graphical icon is self-explanatory. This contributes to a much clearer UI.

- All the icons have the same colour using CSS to maintain a theme. The colour blue is used since it supposedly relaxing to look at and is the popular trend among most websites and apps now days.

- Display.html
  - There are 2 navbars on either side of the page.
  - The left one contains links to logout, home, and bookmarks.
  - The one on the right has elements to add folders and upload files.
  - This layout was chosen so that the user does not have to navigate across different pages to do different things – everything is 'under one roof'. It makes the UI very, very simple since everything is right in front of the user.
  - The central area displays the files and folders along with the absolute path that the user is in. It is kept in the center of the page, so the user naturally focuses on it.
  - When renaming a file, the input text and submit button(that are originally hidden) appear within the table row of the file itself instead of redirecting the user to a separate page.
  - This gives the user a much more natural feel as opposed to being redirected to another page, entering data there, and then being redirected back to the table with the changes displayed.

2. Methods and Modules

- DirectoryHandler.py
  - This file handles most of the user’s requests such as:
    - ❖ Displaying files/folders
    - ❖ Adding new folders
    - ❖ Uploading files
    - ❖ Deleting files/folders
    - ❖ Adding/Removing Bookmarks

Working:

Depending on the value of the button that called the /directoryhandler class, various working takes place.

I. If action==’Add Folder’:
- Adds a new folder to the current parent directory.
- The default directory is the root directory and represented by /
- It reads the name of the new folder to be added and creates an object of the Directory model, and sets its ‘name’ property as the name that the user entered.
- It then adds the ID of this directory to the list of subdirectories of the parent directory.
- Hence in essence, the new folder has been added to the subdirectory.
- The new list of subdirectories is displayed on HTML via a Jinja template

II. If action==’Delete’:
- This bit of code deletes folders.
- NOTE: Directories are only deleted if they do not contain any subfolders or files.
- It takes in the ID of the directory to be deleted, and removes it from the parent directory’s list of subdirectories.(given the condition that it doesn’t itself contain any files or subdirectories)
- It also deletes the object from the ndb datastore.

III. If action==’Open Folder’:
- Used to navigate inside the clicked folder.
- The list of subdirectories and files of the selected folder is appended to lists and displayed in a table in HTML.
- The directory appears to be ‘opened’

IV. If action ==’delete file’
- It deletes files from a folder.
- Since files are represented 2 lists - BlobKeys and filenames, the name of the file to be deleted is sent via a form.
- The current folder’s list of files is searched for the file to be delted.
- The entry at that index is removed from the list of blobs and list of filenames.
- Since this file is no longer needed, it is also removed from the blobstore using the Blob Key.
- The update list of files generated and displayed in HTML using Jinja.

V. If action==’Open Star-folder’:

VI. If action==star-dir’

- This code is briefly discussed under Extra Feature: Bookmarking Folders

VII. If action==’rename’:

- This code is briefly discussed under Extra Feature: rename files

- UploadHandler.py
    - This file handles uploading files.
    - When a user uploads files, it accepts the file and adds it to the datastore.
    - It also checks if a file with the same name exists in that parent directory.
    - If a file with the same name exists, it displays an error message.
    - If not, it adds the file to the datastore and appends the filename in the list of foldernames in the parent directory.

- DownloadHandler.py
    - This folder serves files to the user.
    - Using the Blob key of the required file, it extracts the file from the blob store.
    - The file is sent as a response to the user.

- Login.py
    - This is the landing page of the application.
    - It checks if a user is logged into the application or not.
    - If a user is logging in for the first time, it creates a root directory and StarDirectory for that user.
    - If a user is not currently logged in, and it displays the login page.

Models:

- Directories.py
    - Used to represent a directory in the application.
    - It contains a list of filenames, and BlobKeys to represent files within it.
    - It contains a list of directory IDs to represent subdirectories.
    - It also contains a Boolean list called star_folder to mark whether the folder at the corresponding index is bookmarked or not.

- StarDirectory.py
    - This model is used to represent the bookmark folder.
    - When a folder is bookmarked, it’s ID is appended to a list.
    - The absolute path of the folder is also added to a separate list.
    - When the bookmarks folder is opened, these are used to lists populate the table.

- MyUser.py
  - This model is used to represent users in the application.
  - It keeps track of whether a user has signed in before or not.
  - If a user has logged in, but his details don't exits in this model, that means he is logging in for the first time, and a root directory needs to be made for him.

Extra Feature: Bookmarking Folders

- This feature allows folders/sub-folders to be bookmarked
- Once bookmarked, they will all show up under the 'bookmarked' folder

1. **UI**
   - There is a star displayed beside each folder that is initially only an outline.
   - The user can click on that star to bookmark that folder.
   - Once bookmarked, the star appears filled instead of just an outline.
   - Similarly, clicking a bookmarked star, un-bookmarks the folder.
   - The left-hand navbar contains a link called 'Bookmarks'.
   - On clicking this link, all the bookmarked folders are displayed.
   - When a folder is 'un-bookmarked' it disappears from the bookmarks folder.

2. **Methods and Working**
   - When a folder is bookmarked, an array in the parent directory called star_folder (that marks whether each subdirectory is bookmarked or not) is updated to "True" at the corresponding index of that subdirectory.
   - The ID of this newly bookmarked folder is also added to a list in the StarFolder Model.
   - Conversely, when a bookmark is removed from a folder, its ID is removed from the list in the StarFolder Model, and its corresponding index in the star_folder array in its parent directory is also marked as "False"

Extra Feature: Renaming Files

1. UI

- The edit icon is present in line with files in the table.
- The text input field and submit button are hidden until the edit button is pressed.
- When the edit button, the text input and submit button appear within the table's <TD>
- Ok clicking submit, the changes are reflected in place.
- Hence the user's focus never shifts off the row of the table.
- This provides a very natural feel to the user.

2. WORKING

- The filename to be edited is sent to the /directoryhandler.py
- It checks if a file with the new name already exists
- If it doesn’t, it replaces the name of the file in the list of filenames of the parent directory.
- It then reloads the table in the html with the reflected changes.
- If the filename could not be changed due to an already existing file, an error is displayed in as a popup message in the browser.

***NOTE: while the extension of the file can be changed during renaming, the user can still download and view the file perfectly.**