IT Project Drive

April 29, 2018

1 ITWS Project: Drive

1.1 Team members:

- Ramkishore S
- Antony Martin

1.2 Contents:

1. Implementation
2. Features 1. Console Operations 2. Blogs (inspired from `<username>.github.io`) 3. Shared files
3. Security
4. Installation and Running

1.3 Implementation

- The backend is implemented as a State machine.
- We followed the MVC model.
- File system is used to store User files.

1.3.1 Controllers

- Auth
- Main
- Shared

1.3.2 Models

- User

1.3.3 Views

- main
- login
- signup
- shared_files
- shared_folders
- blog

1.3.4 Controllers

- forms used in controllers can be found in the `forms` folder, one file corresponding to each controller

Auth

- Used for user authentication(login, signup and logout).
- `Flask-Login` is used to manage all authentication tasks.
- `Flask-WTF` is used to manage forms.

Main

- Used for managing user files in the backend File System
- Deals with File system actions like `create dir, delete_file, ls` etc
- Requires user authentication

Shared

- Used for accessing public files

1.3.5 Models

User

- Used to store user data `id, email, username, password-hash`

1.3.6 Views

Main

- For accessing User's data(files and folders),and provides functionalities like `create folder`, `change-folder`, `delete file/folder`, `upload file` etc.

Login

- For user login

Signup

- For user signup

Shared_files

- For displaying public files of particular user

Shared folders

- Displaying public component of every user

Blog

- Each user can upload a file `blog.html` to the root directory which will be served at `/blog/<user id>`
- Inspired by `<username>.github.io` feature in GitHub.
- Currently blogs only support inline styling and scripting, (cannot attach files)

1.4 Features

1.4.1 Console Operations

`cd()`

- `cd("<folder name>")` to change folders. Does not support paths, the folder must be in the current directory
- `cd(".")` to go to root directory of user
- `cd("..")` to go back, if performed from root folder, no change

`ls()`

- To list out files, filesizes and folders in the current directory

`delete_file("<filename>")`

- Deletes the given file/folder is it is present in the current directory
- Does not accept paths, wildcard expressions etc.

1.4.2 Blogs

(Explained in Views)

1.4.3 Shared files

- An User can make a set of files available to everyone, people don't have to be logged in to look at them.
- Currently only files are supported, folders aren't.
- Any file available in the `public` folder available in the `root` directory is public.
- Folders in `public` and any files in them will remain private.

1.5 Security

- The user has only two options at any given time. Perform functions like `upload_file`, `delete_file`, `create dir` etc in the current directory or change directory. The user cannot perform operations from one directory on another, except deleting, where only the immediate children can be deleted.
- This ensures that user cannot execute any malicius commands that might affect the server/other's data
- While this makes it seem like the power of the user is very limited, it does not make much difference with the GUI, because all that will be visible will be the contents of the current directory.
- Many systems like GoogleDrive, OneDrive have similar functionalities.

1.6 Installation and Running

1.6.1 Set environment variables first

```
export FLASK_APP=app
```

If you want to run it in debug mode, set `export FLASK_DEBUG=true`

1.6.2 Now install it

Installation need only be done once `pip install -e`

1.6.3 For running, execute

```
flask run
```