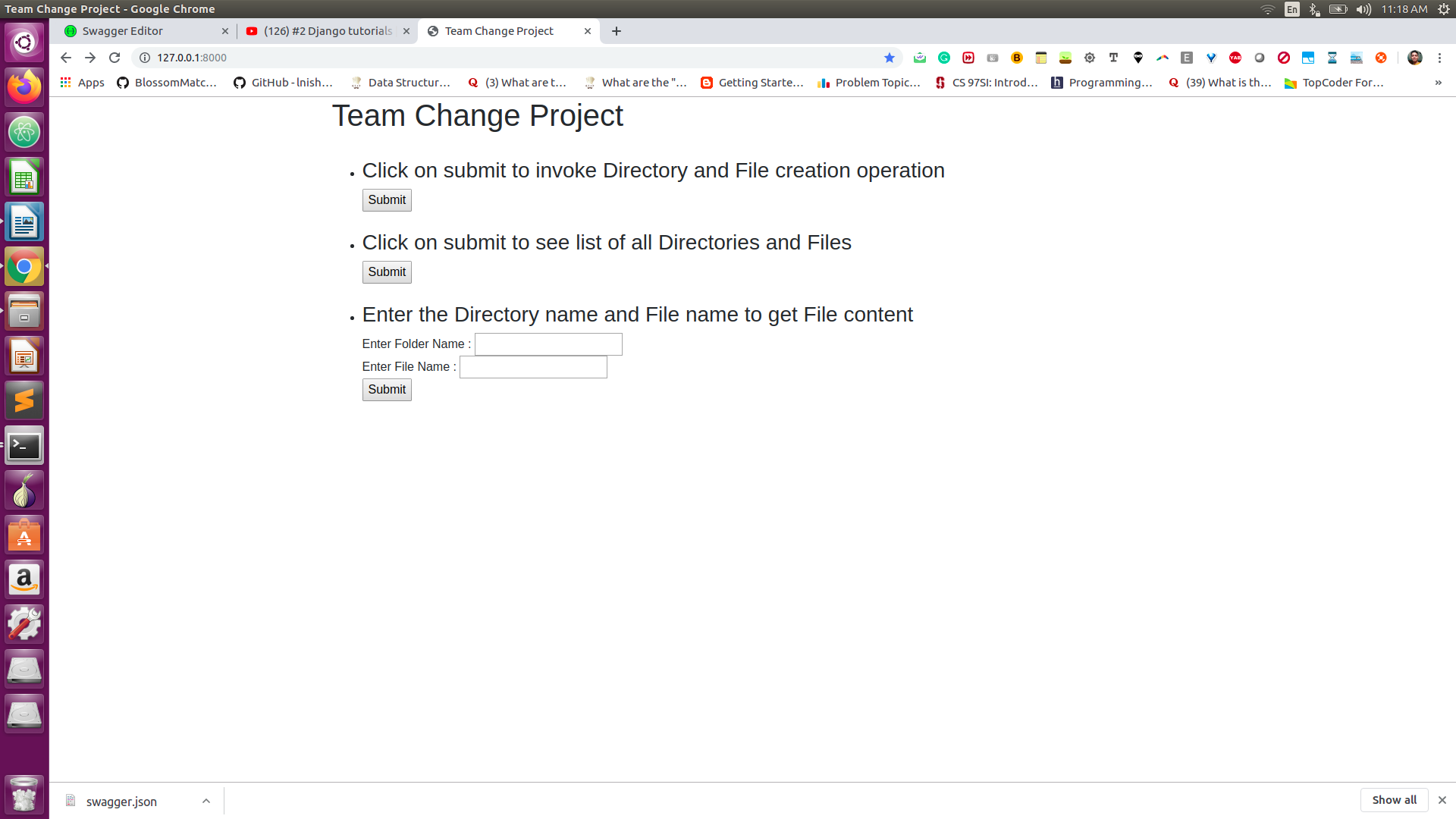
**Team Change Project Code Explanation**

Home Page :

Supports 3 operations on the click of the corresponding submit button :

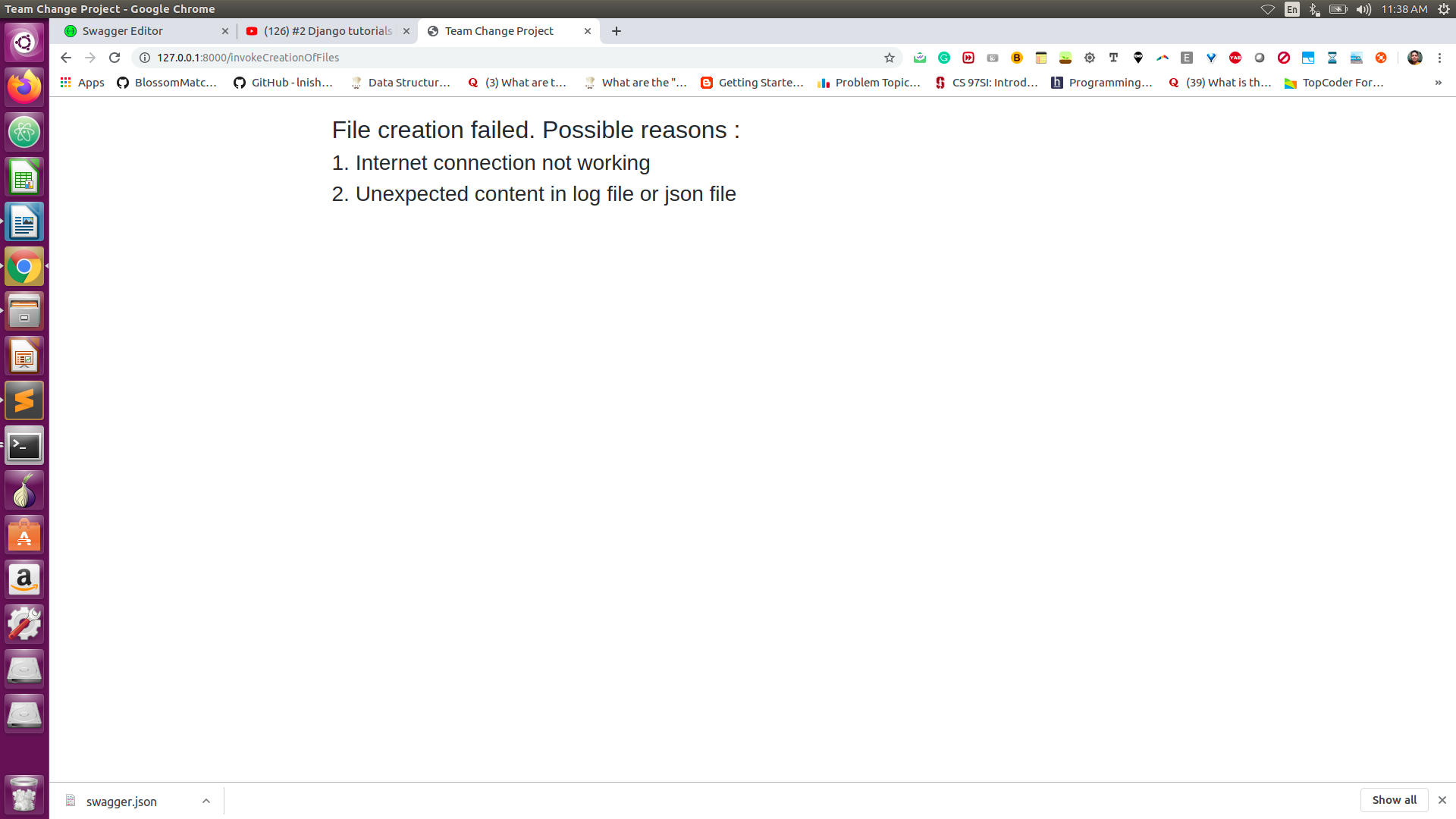
**1. Create directories and files inside the STORAGE\_DIR (value can be set in Processing/config.py) folder.**

**Explanation :**

**-** This function calls createDirectoryAndFiles() function of DoProcessing class in Processing/util\_functions.py .

**-** First the content of the log file and the json file is downloaded using the curl command. This step requires working internet connection otherwise fails and an error is raised which is handled using try catch and website shows an error message and returns **response code 500**.

To check the response code, use Postman API (screenshots in REST API Documentation/Postman)



- If above step is successful, then we go through each line of the log file and store a dictionary named dictIpAddress whose key is ipAddress and value is the list of indexes (line number in log file) on which the ipAddress occurs.

*Time Complexity* : O(N\*L) where N is the number of lines in log file and L is the length of the log line.

*Space Complexity* : O(N) where N is the number of lines in log file and L is the length of the log line.

**-** Then we parse the json file and try to search for each ipAddress. If we find the iPaddress in dictIpAddress, we get the indexes of the log line in which that ipAddress has occured using dictIpAddress and we call getInfoFieldsFromLine( ) function which parses the log line and returns all the relevant information from the line in form of a dictionary.

*Time Complexity* : O(M\*L) where M is the number of ipAddress in json file.

*Space Complexity* : O(L) where L is the length of the log line.

**-** Also, we make a finalContentDict whose key is (pair of directoryName and fileName) and value is a list of (time +’ ’+message) (appended together) from the log line. If we encounter an exisiting key with same directoryName and fileName, we append its (time +’ ’+message) to the end of the list.

*Time Complexity* : O(M\*L) where M is the number of entries in json file.

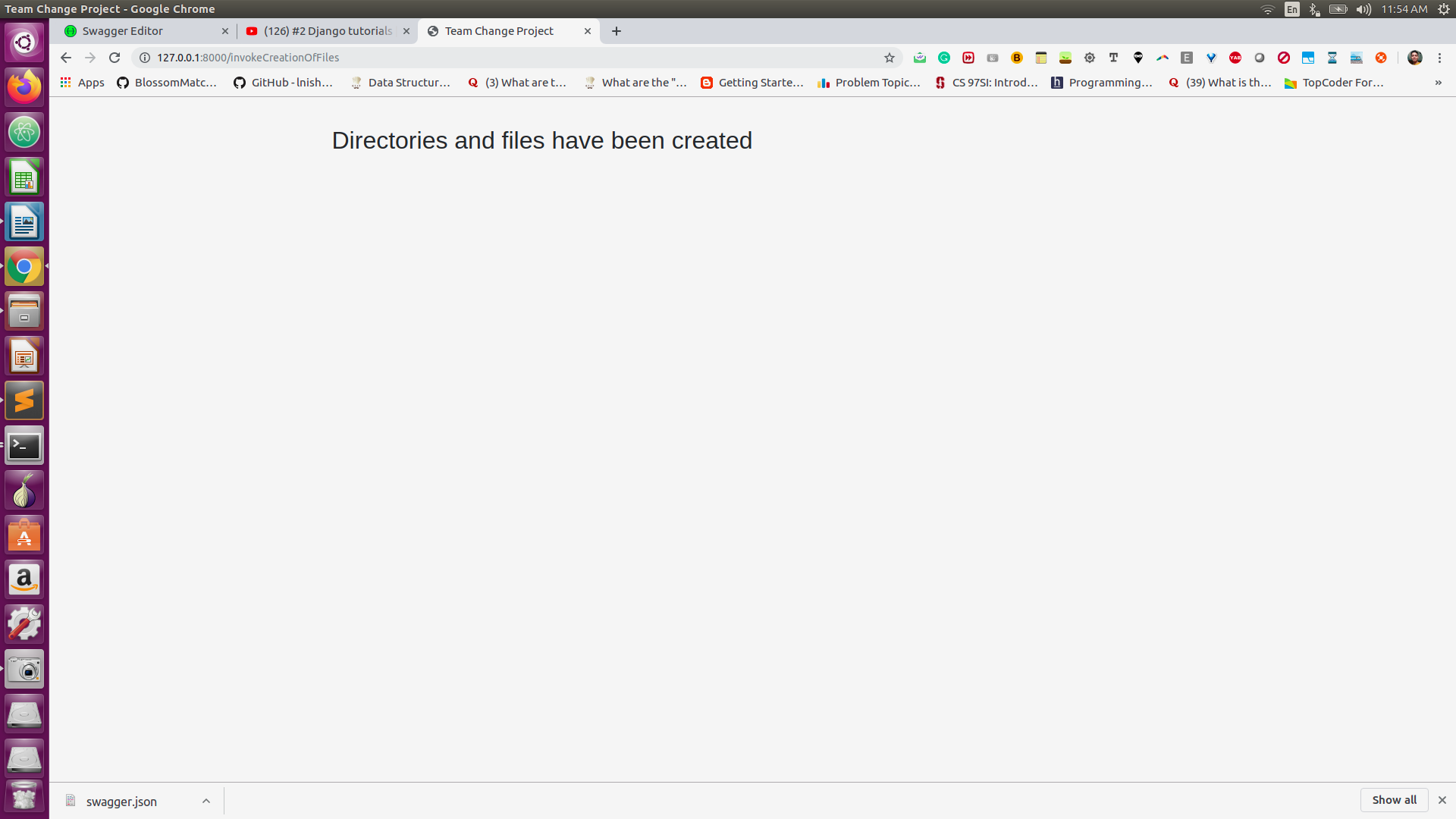
*Space Complexity* : O(N\*L) where N is the number of entries in log file.

**-** After above step, we iterate through the dictionary and make directories and files with the appended (time+’ ’+message) as its content. We overwrite if the file already exists.

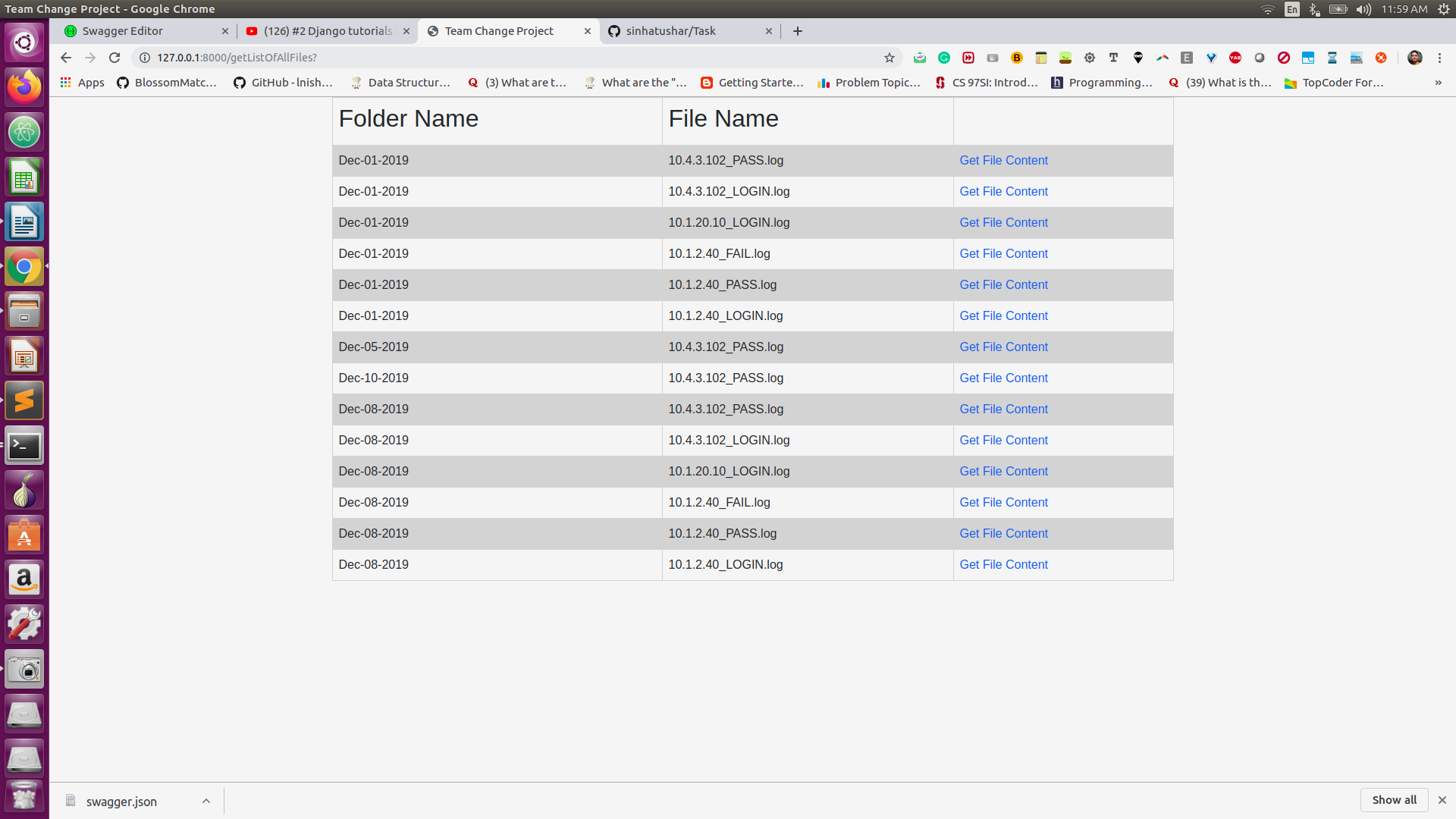
*Time Complexity* : O(N) where N is the number of entries in log file.

*Space Complexity* : O(N\*L) where N is the number of entries in log file.

On success, **response code of 200** is returned with a success message.



**2.** **Shows List of all the files inside STORAGE\_DIR .**



**Explanation :**

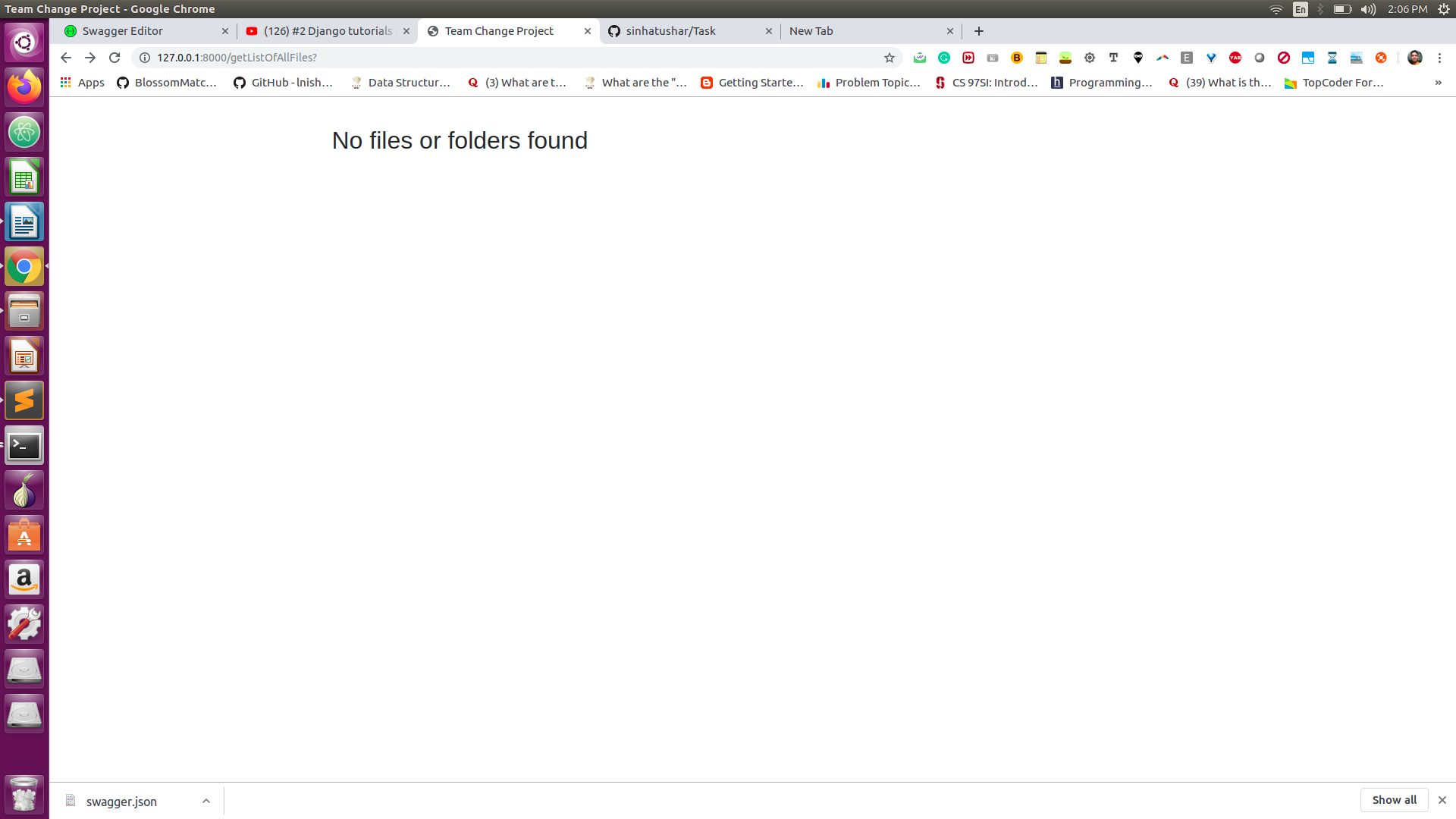
- Calls getContentForAll() function of FindContentForAll class in Processing/util\_functions .py . This function makes use of os.listdir() method to get the name of all the folders and files present in the STORAGE\_DIR directory.

*Time Complexity* : O(N) where N is the number of entries in log file.

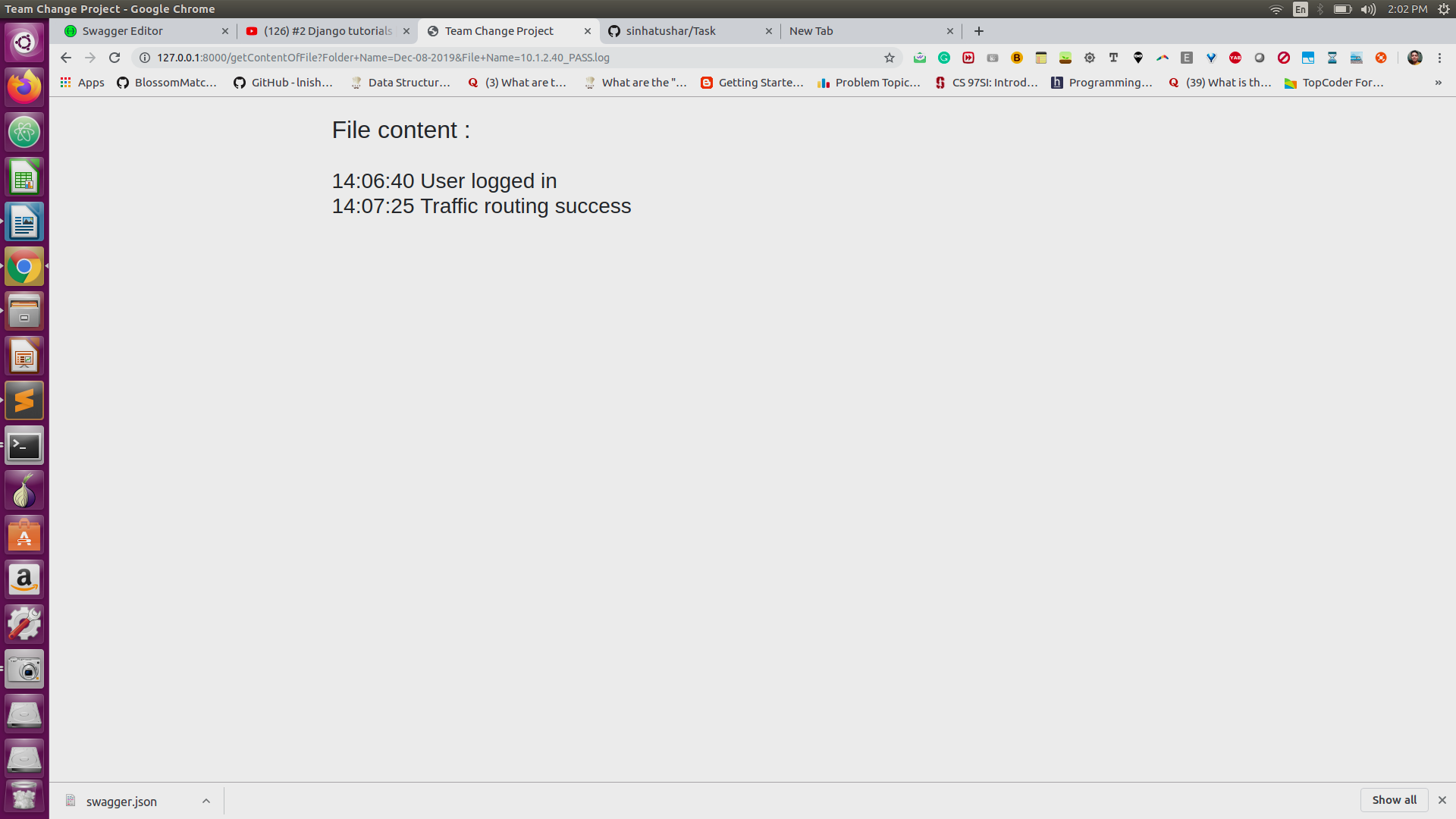
*Space Complexity* : O(1)

- The name of the folder and the file are stored in the dictionary named contentDict and the the content inside the file is obtained using getFileContent() function of FindContentForFile() class whose functioning is explanied next. This contentDict is appended to a list named contentDictList which is returned when all the files have been covered.

- If there are no files stored in STORAGE\_DIR, then the length of contentDictList is 0 and we return a **response code of 404** indicating that there is no file found in the storage.



**3 . Show the content of a file name if the parent folder name and file name are supplied as aruguments.**

****

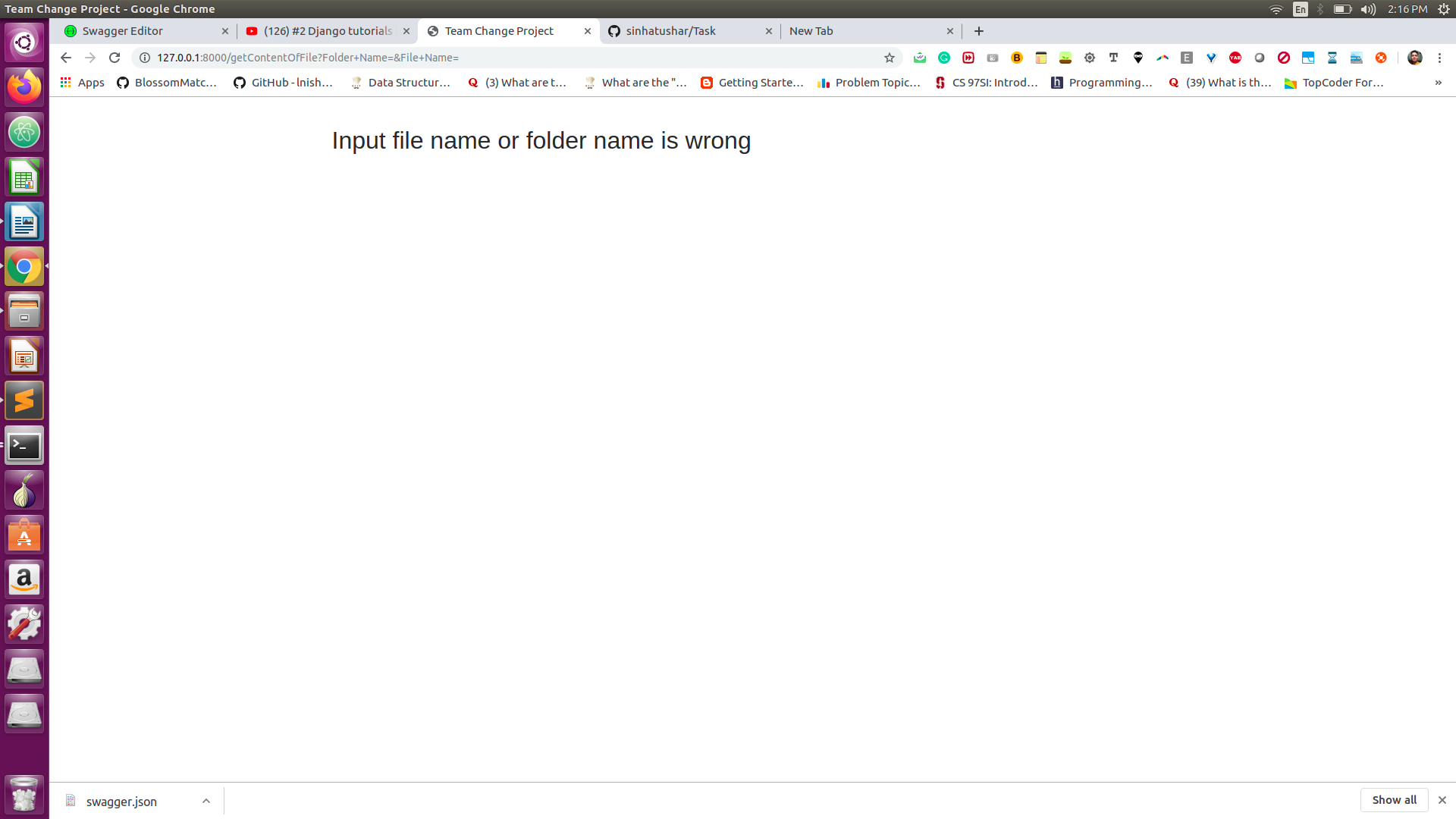
**Explanation :**

**-** Calls getContentOfFile( ) function of FindContentForFile class in Processing/util\_functions.py . This function reads the file and returns the content if the parent folder name and file name are correct.

If the file name and folder name are not correct, it raises a FileNotFoundError and returns a **response code of 400** as an indicative of wrong input.

*Time Complexity* : O(L) where L is the length of the log line.

*Space Complexity* : O(L) where L is the length of the log line.

******