The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously.
Please read all guidelines carefully before starting the review.
- Please, stay courteous, polite, respectful and constructive in all communications during t
his review.
- Highlight possible malfunctions of the work done by the person and take the time to disc
uss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and t
he scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction bef
ore starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an “empty work” if repository do
es not contain the work of the student (or group) in the src folder of the develop branch, or
“cheat” in case of cheating or if the student (or group) are unable to explain their work at a
ny time during review as well as if one of the points below is not met. However, except for
cheating cases, you are encouraged to continue reviewing the project to identify the probl
ems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the grou
p.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of t
he check. The final decision on project evaluation remains with the reviewer.

MAIN PART

Exercise 00 - Fool Me Once

Check that "credentials.py" is present
Check that the webserver is started on port 8888 when script is launched
Check that the webserver uses WGSI interface (function with two arguments) returning an i
terator over response body
Check that response has "Content-Type" set to "application/json"
Check that HTTP server returns appropriate credentials for the species from the known list
Check that for known species server HTTP response status code is 200
Check that HTTP server returns `{"credentials": "Unknown"}` for the species not in the kn
own list
Check that for unknown species server HTTP response status code is 404

No Yes

Exercise 01 - Screwdriver Song

Check that `README` file is present and contains the command on how to run the test HTTP server
Check that both server code (may contain multiple files) and client script `screwdriver.py` are present
Check that webserver listens on port 8888
Check that the webserver allows uploading files using a browser through the main page
Check that the webpage shows the list of all the audio files already uploaded
Check that the webpage shows an error "Non-audio file detected" when uploaded an image (like JPG), text or other non-audio content
Check that non-audio files are discarded (not saved on disk) and not shown in a list of uploaded files
Check that the client script allows uploading file to the server via HTTP using syntax like `python screwdriver.py upload /path/to/file.mp3`
Check that the client requests a list of uploaded files from the server and prints it when called like `python screwdriver.py list`

No Yes

Exercise 02 - Good Timing

Check that "doctors.py" script is present
Check that the script actually uses threading (either library `threading` or `concurrent.futures`) and threads are competing for the resources while determining the order
Check that the script launched several times produces an output like ``` Doctor 11: BLAST! Doctor 9: BLAST! Doctor 12: BLAST! Doctor 10: BLAST! Doctor 13: BLAST! ``` where the order of lines might be different, but there should be exactly five of them every time with no doubles.

No Yes

BONUS PART

Bonus section for EX01

Check that uploaded audio files can actually be played in the browser from the main webpa