

# Python project

March 2020

Complete the two following projects. Beside the specified details, you are free to design them and structure your code as you wish. Please, comment your choices and the general idea you used in a *README.txt* file. Upload an archive containing all necessary files to run your project, along with the *README.txt* file, to the MOODLE platform **by the end of March**.

**NOTE: the projects have to be implemented individually.**

---

## URL Shortener

URLs can be extremely long and not user-friendly: they can be filled with difficult characters and don't form meaningful words, making thus difficult to remember them. A URL Shortener reduces the characters or letters in a URL, making them easier to read and remember. For instance, a URL like *xyz.com/wwryb78&svnhkn%sgqh?sfiyh* can be shortened to *xyz.com/piojwr*

Design and write a Python application to shorten URLs and redirect users to the original URL when the shortened URL is visited. In the application, the users will input the original URL, and they will get the new, shortened URL as result. To do this, you can use a combination of the **random** and **string** modules to generate the characters for the shortened URL. *Describe the algorithm you used to shorten the URL in a short comment at the beginning of your code.*

You need to save the original and shortened URLs in a database, that for the purpose of this project can be implemented by using a dictionary. When a request comes in, the application checks if the URL exists and redirects to the original URL, or else it redirects to a 404 page.

To implement the redirection, you can just print a string text that outputs the result of the operation.

**EXTRA:** To implement the redirection, use an actual library that opens a URL, such as **urllib**.

---

## Quiz Application

The Quiz Application will present questions to the users and expect the right answers to those questions.

The main objective of this project is to set quizzes and have people answer them. Therefore, users should be able to set questions, and other users should be able to answer those questions. The application will then display the final score and the right answers.

Users creating the tests should be able to create tests with the questions and answers by simply uploading a text file. The text file will have **a format that you can decide**, so the application can convert from a file to a quiz.

You need to implement a database for this project, that for the purpose of this project can be implemented by using a dictionary. The database will store the questions, possible answers, correct answers, and the scores.

Once the database has been set up, a user can take the quiz: a set of 10 questions is presented to the user, randomly selected from the ones stored in db. At the end, the user gets the final score.

*Design the solution as you wish, and describe your approach in a short comment at the beginning of your code.*