

מסמך דרישות

שם פרויקט: VPL

מס' מזהה	תיאור	סוג הרשאה/ משתמש	סוג (פונקציונאלית/ לא פונקציונאלית)	הערות
1	Upload new assignment with VPL plugin	Practitioners/Lecturers /assignment checkers	Functional	Below
2	Upload execution files (scripts) for the pre-running	Practitioners/Lecturers /assignment checkers	Functional	
3	Submit the solution multiple time and receive feedback on the work in real-time	Student	Functional	
4	Review the student submissions	Practitioners/Lecturers /assignment checkers	Functional	
5	Submit grade for each student considering the proposed grade and the other properties	Practitioners/Lecturers /assignment checkers	Functional	
6	Support variety of programming languages	VPL plugin	Functional	



Explanation:

The VPL plugin must be installed and configured on the Moodle platform.

- * The plugin should allow educators to create and upload test cases for each programming task, including the expected input and output for each test case.

Part of the tests will be pulled by a script into the VPL plugin, and store them in the workspace environment location.

These test files may be uploaded by the lecturer or practitioner either through GitHub or manually.

The VPL tests should be dependent on these test files.

- * Students should be able to submit their code multiple time and receive feedback on their work in real-time.

The submission will be due to the specific programming course requirements (ZIP folder and GitHub URL etc.)

- * The plugin should allow the educators to review the student submissions

according to pre-written script which we will provide.

The script should be able to accept a GitHub URL or a ZIP file as an input.

- If a GitHub URL is provided, the script should clone the repository and extract the necessary files .

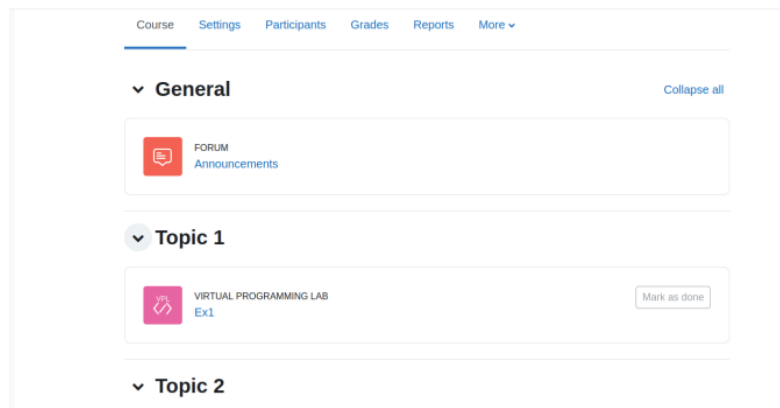
- If a ZIP file is provided, the script should extract the necessary files from the ZIP file.

The extracted files will be saved in an appropriate location in the working environment.

- * The plugin should be integrated with the Moodle gradebook, allowing educators to easily assign grades to students based on their code submissions.

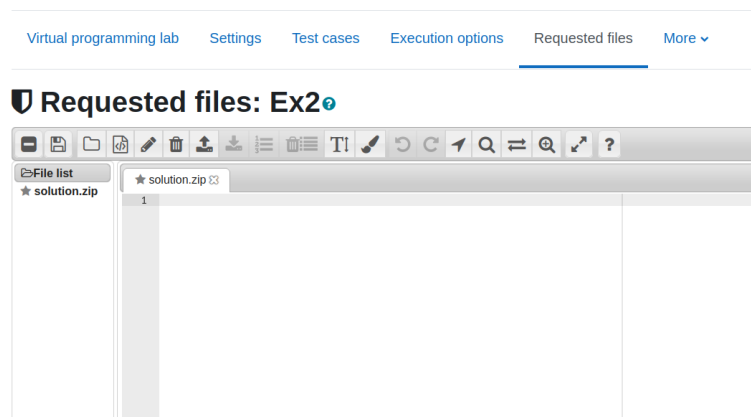
The procedure:

1- open new assignment with VPL plugin:



2- In the setting for the new assignment we have few options for the students submission.

For the courses that the students submit their code with ZIP directory (C, Networking, Introduction to Programming)- the practitioner will create a new file in the requested files tab, with .zip extension



The students will drag the file to :

Programming languages / Ex2

VPL VIRTUAL PROGRAMMING LAB

Ex2

Description Submission </> Edit Submission view

▼ **Submission**

Comments

solution.zip

Choose a file... Maximum size for new files: 2 MB

You can drag and drop files here to add them.

For the courses that the students submit their code with git URL (C++, OOP, Operating System, DataBases), the practitioner will create a new file in the requested files tab, with .txt extension.

Virtual programming lab Settings Test cases Execution options Requested files More ▼


Requested files: Ex1

★ GitLink.txt

1

The students will drag the file to :

Programming languages / Ex1

 VIRTUAL PROGRAMMING LAB


Ex1

[Description](#) [Submission](#) [Edit](#) [Submission view](#)

▼ **Submission**

Comments

GitLink.txt Maximum size for new files: 2 MB



You can drag and drop files here to add them.

3- For those two options, practitioners will add the pre-run-script to the execution file. It will be the same script for both types of submission.

As well this script will test the student's submission. Example:








</> Execution files: Ex2

```












vpl_run.sh vpl_debug.sh vpl_evaluate.sh vpl_evaluate.cases pre_vpl_run.sh
1 #!/bin/bash
2 echo "after ls -l"
3 echo "-----"
4 unzip Notebook-a.zip
5 echo "after unzip"
6 echo "-----"
7 cd Notebook-a
8 echo "after cd"
9 echo "-----"
10
11 chmod +x grade
12 echo "-----"
13 ./grade | grep "Grade: 100"

```

4- Practitioners/Lecturers/assignment checkers get proposed grade from VPL evaluation , that they will be able to submit for the student:

		name 					
		Surname 	Submitted on 	Submissions 	Grade 	Evaluator 	Evaluated on 
1	ST	Student Tal	Saturday, 17 December 2022, 1:42 PM	1	Proposed grade: 0 / 100		

VPL

		First name 					
		Surname 	Submitted on 	Submissions 	Grade 	Evaluator 	Evaluated on 
1	ST	Student Tal	Sunday, 27 November 2022, 7:03 PM	1	Proposed grade: 0 / 100		<div>  Submission view  Previous submissions list  Grade  Copy </div>

VPL

After teacher check :

		First name ↓ / Surname ↓	Submitted on ↓	Submissions ↓	Grade ↓	Evaluator ↓	Evaluated on ↓	⚙️ ↓
1	ST	Student Tal	Sunday, 27 November 2022, 7:03 PM	1	100.00 / 100.00 (0.00000)	Teacher Teacher	Monday, 12 December 2022, 11:57 AM	⚙️ ↓

The practitioners will have as well an option to choose for automatic grade, which will be proposed by VPL when the student submit his code and it will be updated directly on the student's moodle.

		First name ↓ / Surname ↓	Submitted on ↓	Submissions ↓	Grade ↓	Evaluator ↓	Evaluated on ↓	⚙️ ↓
1	SS	student stud	Wednesday, 30 November 2022, 4:44 AM	7	100.00 / 100.00	Automatic grade	Monday, 26 December 2022, 7:45 PM	⚙️ ↓

#	First name / Surname	Grade
1	Automatic grade	1/1 (100.00%)

5- From student side : Student can submit their code as many time as they want, and get a grade due to Practitioner's test:

Grade

Reviewed on Monday, 12 December 2022, 11:57 AM by Teacher Teacher

Grade: 100.00 / 100.00

Assessment report 📄 [-]

[\[+\]Summary of tests](#)