BASH SCRIPTING IN VPL

MINOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

Information Technology

Submitted By:	Submitted To:
Rudra Singh (1805549)	Prof. Harjot Kaur Gill
Tanisha Gupta (1805461)	Assistant Professor
Saurav Kumar (1805449)	Minor Project Coordinator

Department of Information Technology

Guru Nanak Dev Engineering College,

Ludhiana-141006

Abstract

In this project we will go step-by-step through the whole process of setting up moodle , building and configuring programming assignments using VPL: - beginning at the simplest programming exercise, where VPL acts simply as an homogeneous, Moodle-integrated programming and running environment and allows the teacher to plan and perform both lab activities that require student attendance and assignments for distance courses, without worrying if the required IDE, compiler or operating system version is properly installed at the lab or the student's home, - up to complex programming projects where VPL controls a set of source code files, some provided by the teacher and some developed by the student, acts as coding style and plagiarism judge and automatically evaluates and grades the projects, running tests based upon a set of test cases developed by the teacher.

ACKNOWLEDGEMENT

WE are highly grateful to the Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the minor project work at BASH SCRIPTING IN VPL.

The constant guidance and encouragement received from Dr. Kulvinder Singh Mann, H.O.D., IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

WE would like to express a deep sense of gratitude and thanks profusely to Prof. Ranjodh Kaur, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner.

WE express gratitude to other faculty members of Information Technology Department of GNDEC for their intellectual support throughout the course of this work.

Finally, WE are indebted to all whosoever have contributed in this report work.

Rudra Singh

Tanisha Gupta

Saurav Kumar

List of Figures

Title page

Contents

1 Introduction

1.1 Introduction to Project Page

This report illustrates that how to setup a VPL - Virtual Programming Lab module for Moodle activity that tests a Bash script that takes as input an integer and acts as a teller machine. Moodle is a learning platform designed to provide educators, administrators and learners with a single robust, secure and integrated system to create personalized learning environment. Achievement of computer programming skills requires a lot of training by means of real program-development assignments. Managing and assess the students submissions for those assignments could be a very complex task. Availability of tools to organize the assignments, receive and storage the submissions, support automatic or semi-automatic assessment and provide feedback could be very helpful.

1.2 Project Category

- Internet based
- Application or System Development
- System Administration)

1.3 Objectives

1. To setup script for auto evaluation capabilities.
2. To facilitate the preparation of assignments, manage the submissions.
3. To use a particular programming language to aid a powerful and flexible assessment tools based on program testing.

1.4 Identification/Reorganization of Need

Integration of this kind of tools into a Learning Management System is an essential feature in order to improve students performance.

- To provide a very simple development environment in order to smooth the learning curve to the students.
- To be independent of the programming language.
- To use a particular programming language it is only required that the appropriate compiler is installed which is not required in virtual programming lab environment by moodle.

1.5 Existing System

VPL is the easy way to manage programming assignments in Moodle. Its features of editing, running and evaluation of programs makes learning process for students, and the evaluation task for teachers, easier than ever.

1.6 Proposed System

Number of submissions performed by a student is currently not limited, we believe such a feature would be beneficial. We have observed students using Eclipse to generate their Java program, submit their first draft to VPL, and then stay in the VPL edit window to finish up getting rid of the final bugs. The due date and time of an assignment can be recorded in the test script, allowing the script to lower the grade by some selected amount for every day past the deadline.

1.7 Unique Features of the System

1. The instructor can define the rubric under which a VPL grade is assigned. This is controlled per VPL-assignment.

2. The instructor can make the grade visible to the student, or not. In the latter case, the grade is revealed after the due date.

3. The instructor can controls the resources needed by the jail server.

4. For a given VPL activity, the programs submitted can be those of an individual student, or from a group of students.

5. The instructor can enforce for programs to be typed by hand in the submit window, and disable copy/paste of program code.

6. The instructor defines how the student program is evaluated and graded. This allows for testing properties of a program other than its.

2 Requirement Analysis and System Specification

2.1 Feasibility Study

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new device and debugging old running device/system. All system is feasible if they are unlimited resources and in nite time. There are aspects in the feasibility study portion of the preliminary investigation:

2.1.1 Technical Feasibility:-

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to work properly?
- Are there technical guarantees of accuracy, reliability, case of access and control?

2.1.2 Operational Feasibility:-

Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there controlled efficiently?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undetermined the possible application benefits.

2.1.3 Economical Feasibility:-

In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

2.2 Software Requirement Specification Document which must include the following:

Minimum software required:-

- Win 10 or Ubuntu 18.04
- Xampp

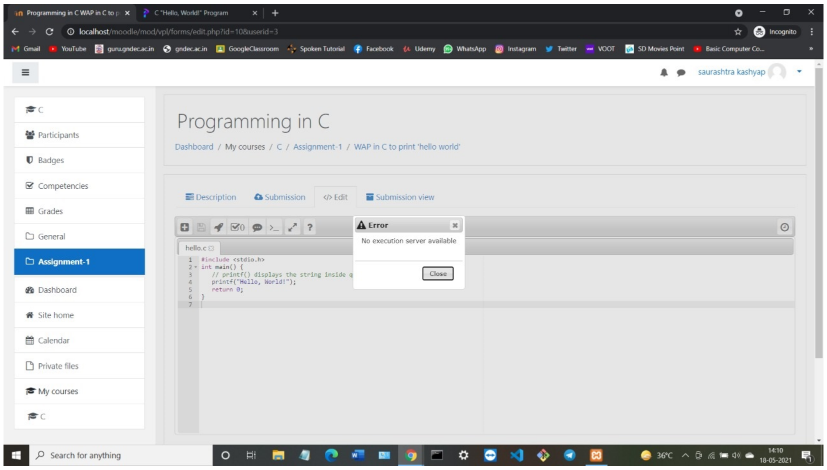
- Moodle
- VPL Plugin

Minimum hardware required:-

- Intel dual core i5
- 512 GB Hard disk
- 12 GB RAM

2.3 Expected hurdles

Connection errors of apache and mysql server with localhost:-



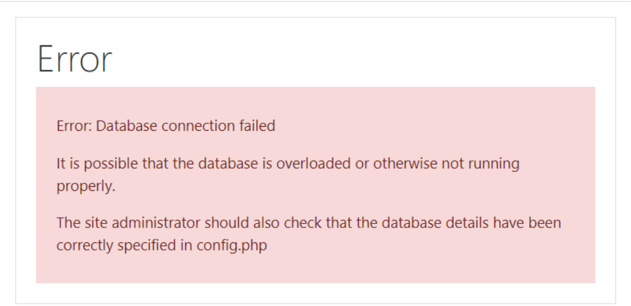
```
#include <stdio.h>
int main() {
   // printf() displays the string inside q
   printf("Hello, World!");
   return 0;
}
```

Error

No execution server available

Figure 1: Execution Error

Moodle updates cause connection errors:-



Error

Error: Database connection failed

It is possible that the database is overloaded or otherwise not running properly.

The site administrator should also check that the database details have been correctly specified in config.php

Figure 2: Moodle Database Error

3 System Design

3.1 Design Approach

The design approach of this project is Object Oriented.

3.2 Detail Design

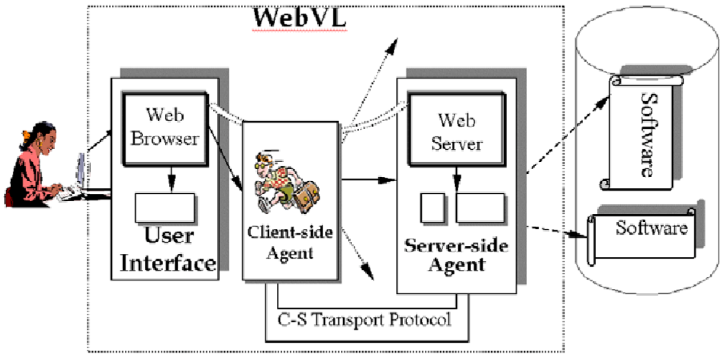


Figure 3: Virtual Programming Lab Architecture

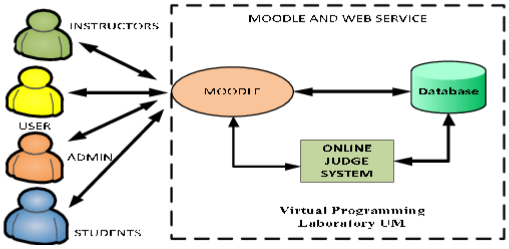


Figure 4: Virtual Programming Lab UM

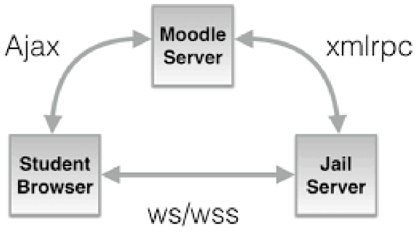
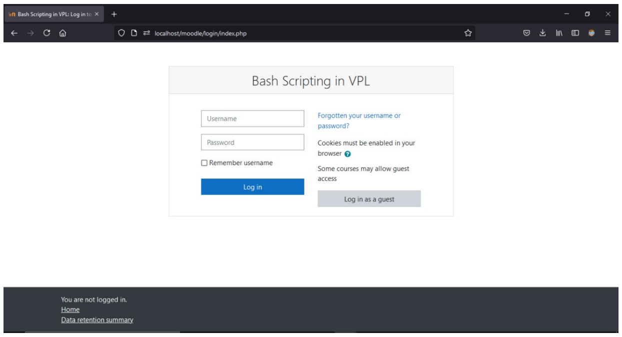


Figure 5: Interconnected Moodle Servers

3.3 User Interface Design

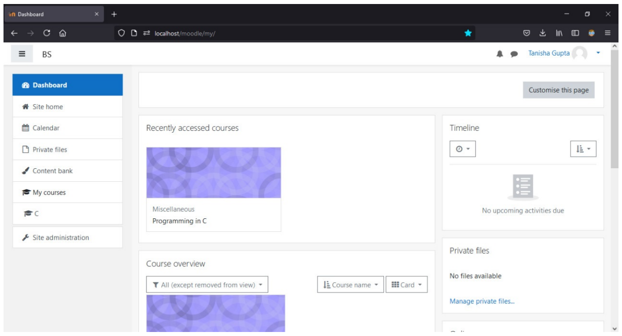


Cookies must be enabled in your browser

Some courses may allow guest access

You are not logged in.

Figure 6: Default Login view of Moodle



No upcoming activities due

No files available

Figure 7: Example of the Image

3.4 Database Design

3.4.1 ER Diagrams

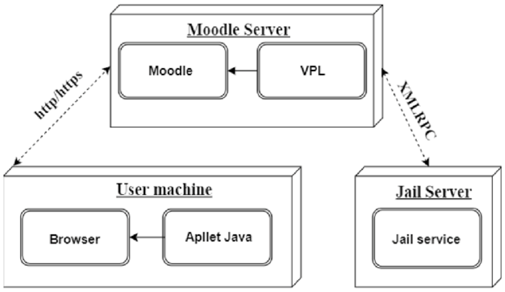


Figure 8: Moodle Server

3.4.2 Database Manipulation

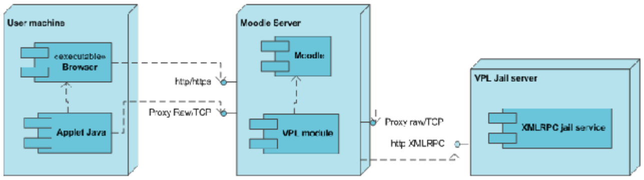


Figure 9: Detailed Server view of VPL in moodle

3.4.3 Database Connection Controls and Strings

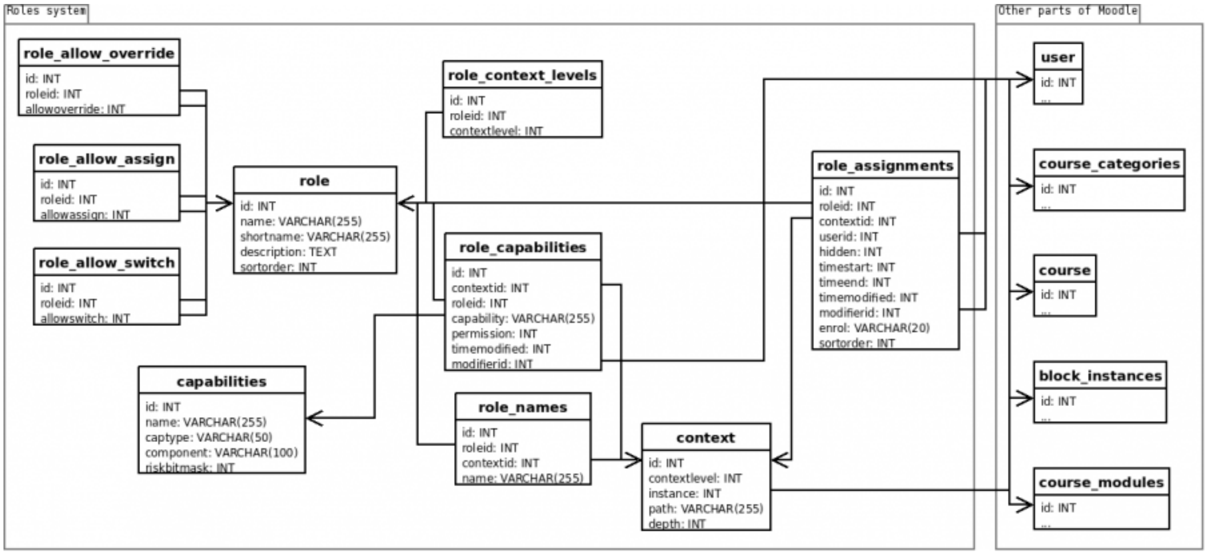
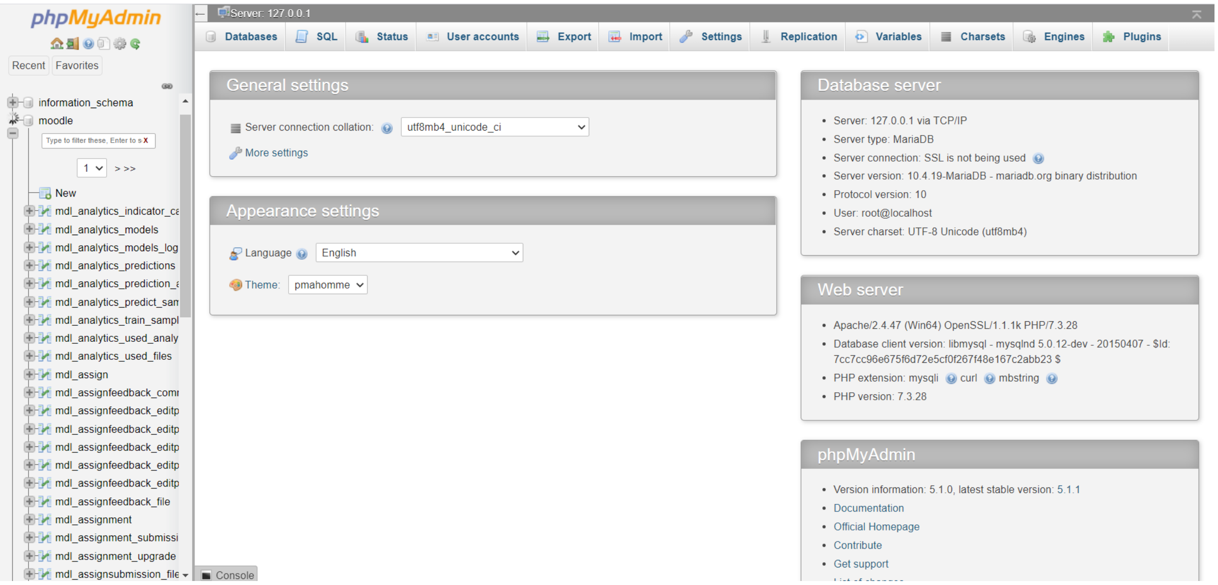


Figure 10: Database Connection controls

3.5 Database Design



General settings

Server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.19-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.47 (Win64) OpenSSL/1.1.1k PHP/7.3.28
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 $
- PHP extension: mysqli curl mbstring
- PHP version: 7.3.28

phpMyAdmin

- Version information: 5.1.0, latest stable version: 5.1.1
- Documentation
- Official Homepage
- Contribute
- Get support

Figure 11: Moodle Database design

3.6 Methodology

Based on VPL-Virtual Programming Lab, it is a activity module that manage programming assignments and whose salient features are:-

- Enable to edit the programs source code in the browser.
- Students can run interactively programs in the browser.
- You can run tests to review the programs.
- Allows searching for similarity between files.
- Allows setting editing restrictions and avoiding external text pasting.

4 Implementation, Testing, and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

4.1.1 Introduction to Languages:-

Bash:- Bash is a command language interpreter. It is widely available on various operating systems and is a default command interpreter on most GNU/Linux systems.

The name is an acronym for the 'Bourne-Again SHell'.

Shell:- Shell is a macro processor which allows for an interactive or non-interactive command execution.

Scripting:- Scripting allows for an automatic commands execution that would otherwise be executed interactively one-by-one.

4.1.2 Tools and Technology used:-

- Win 10 or Ubuntu 18.04
- Xampp
- Moodle
- VPL Plugin

4.2 Testing Techniques and Test Plans

Teller.sh:-

```
#! /bin/bash
amount=$1
no20s=$( expr $amount / 20 ) amount=$( expr $amount % 20 ) no10s=$( expr $amount / 10 ) amount=$( expr $amount % 10 ) no5s=$( expr $amount / 5 ) no1s=$( expr $amount % 5 )
echo "" echo $no20s echo $no10s echo $no5s echo $no1s
```

Vpl Run:-

```
#! /bin/bash
cat > vpl_execution <<EOF #! /bin/bash
chmod a+rx teller.sh
read var1 ./teller.sh \${var1}
EOF
```

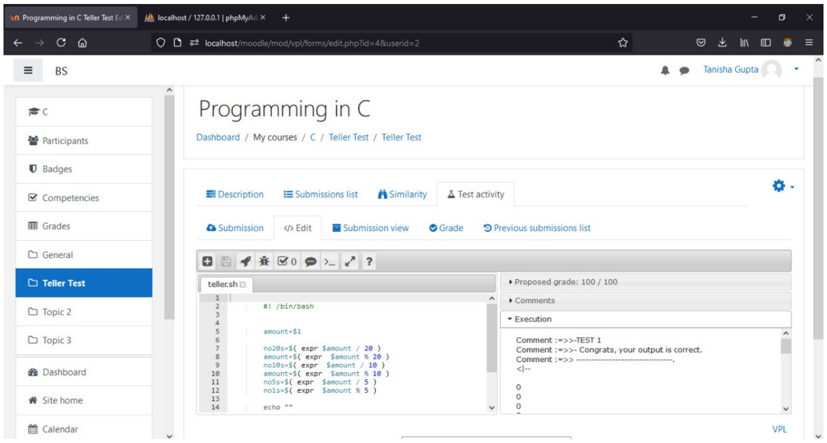
```
chmod +x vpl_execution
```

Vpl Evaluate:-

```
#! /bin/bash
cat > vpl_execution << 'EOF'
#! /bin/bash
javac -J-Xmx128m VPLFakeTester.java
javac -J-Xmx128m VPLJavaTester.java java
VPLJavaTester Hw1_1.java patterns.txt
EOF
chmod +x vpl_execution
```

5 Results and Discussions

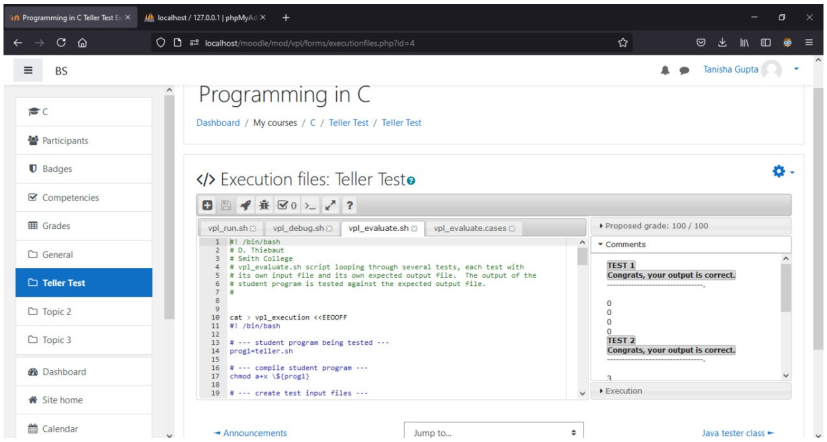
The number of submissions performed by a student is currently not limited, we believe such a feature would be beneficial. We have observed students using Eclipse to generate their Java program, submit their first draft to VPL, and then stay in the VPL edit window to finish up getting rid of the final bugs. The due date and time of an assignment can be recorded in the test script, allowing the script to lower the grade by some selected amount for every day past the deadline.



```
#! /bin/bash


amount=$1

no20s=$( expr $amount / 20 )
amount=$( expr  $amount % 20 )
no10s=$( expr  $amount / 10 )
amount=$( expr  $amount % 10 )
no5s=$( expr  $amount / 5 )
no1s=$( expr  $amount % 5 )

echo ""
```

```
Comment :=>>-TEST 1
Comment :=>>- Congrats, your output is correct.
Comment :=>> ------------------------------.
<|--

0
0
0
```

Figure 12: Output of Teller.sh



```
#! /bin/bash
# D. Thiebaut
# Smith College
# vpl_evaluate.sh script looping through several tests, each test with
# its own input file and its own expected output file.  The output of the
# student program is tested against the expected output file.
#


cat > vpl_execution <<EEOOFF
#! /bin/bash

# --- student program being tested ---
prog1=teller.sh

# --- compile student program ---
chmod a+x \${prog1}

# --- create test input files ---
```

```
TEST 1
Congrats, your output is correct.
------------------------------.

0
0
0
0
TEST 2
Congrats, your output is correct.
------------------------------.

3
```

Figure 13: Output of vpl_evaluate.sh

6 Conclusion and Future Scope

The fast increase we see in CS departments nationwide require changing the way we assess programming skills. We have chosen to implement the automated evaluation of student programs in three of our heavily enrolled classes. Our early experience with VPL is positive. The wide array of programming languages VPL supports, its robustness of implementation, and the flexibility it offers compensate for its complexity of use, and its currently sparse documentation. We have started releasing scripts we have generated for various assignment in an effort to share our experience, ideas, and solutions, hoping others can benefit from our experiment.

A major feature of VPL is its capacity to produce complete assessment reports based on program testing. Moreover the required program tests can be configured in a very flexible way, ranging from simple input-output tests to complex combinations of unit tests, coverage tests or style tests. Another important feature of VPL is the embedded tool to check submissions . It is important because managing large student data is a big problem in academia, so vpl and moodle will make it easy.

References/Bibliography

[1] About Moodle . [Online]. Available: https://docs.moodle.org/310/en/About_Moodle, 2020.

[2] Juan C. Rodríguez-del-Pino, Enrique Rubio-Royo, Zenón J. Hernández-Figueroa. (2012). A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features, [Online]. Available: https://www.researchgate.net/publication/275652921_A_Virtual_Programming_Lab_for_Moodle_with_aut

[3] Virtual Programming Lab for Moodle . [Online]. Available: https://moodle.org/plugins/mod_vpl, 2020.