

Vision Statement

AYU project:

Project goals –

- our main goal is to enable Moodle academic users to submit a moodle assignments with an automatic interface testers designed by the academic stuff
- to create a plug-in that will enable to connect between the moodle system and "Badkan" website which is an automatic tester.

Project scope –

- the plug-in will be design to work with an automatic interface tester "badkan" website
- the plug-in may support connection with other systems such the "Badkan" in this project scope.

High-level features –

- assignments will upload via moodle and will connect to Badkan in the submission page by clicking on a unique button.

Major milestones and deliverables –

- according to project time table (can be found at www.ariel.zone).
- Learning PHP and JavaScript
- Learn how to work with the moodle platform and how does it works
- Learn the "badkan" system and how to work with it
- Join the moodle developers community

Elevator statement –

For moodle system users who use the badkan website and want to improve the experience of task submissions and grade feedbacks.

The AYU is a moodle plug-in that gives the moddle users best submission and feedback expeirence to programing tasks. Unlike the current situation that users need to submit both to the moodle and the Badkan and then the educators need to manually assign grade to each and every student separately ,our product allows users to get both of those actions at once and get the grade automatically and conveniently.

Market survey-

Used 'moodle plugin automatic code grader' search word at google

1. "CodeRunner".

"CodeRunner" is a free open-source question-type plug-in for Moodle that can run program code submitted by students in answer to a wide range of programming questions in many different languages. It is intended primarily for use in computer programming courses although it can be used to grade any question for which the answer is text. It is normally used in Moodle's *adaptive* quiz mode; students paste in their code in answer to each programming question and get to see their test-case results immediately. They can then correct their code and resubmit, typically for a small penalty.

2. Plagiarism plugin.

This plugin uses JPlag and MOSS as the underlying detection engines to provide an educational tool to raise awareness of students on code plagiarism. It permits

Quick and easy scanning of assignments, students' draft submission, easy report navigation, multiple scanning and history, marking pairs of students with possible plagiarism.

3. CodeGrade.

Codegrade is a blended learning application designed especially for programming education. It makes grading and handing in programming exercises more insightful and effective for students and grading more efficient and intuitive for teachers. This is achieved by providing an online environment carefully designed to suit the needs of modern programming education.

It provides:

Inline feedback, making it possible to write feedback on specific parts or lines of code.

Rubrics, easy to use and fully integrated in the Code Viewer.

Rubrics aid consistency between teachers and help students understand their grade.

Automatic linter feedback, feedback of linters is automatically parsed and displayed within the Code Viewer.

All the above are really good tools for the educator to use.

But, our plugin will connect the student to the Badkan code runner.

As a result, in case of mistakes and issues with the student code, the student will get a chance to current its mistakes and learn better. We don't want to fail the students rather to improve their coding skills.