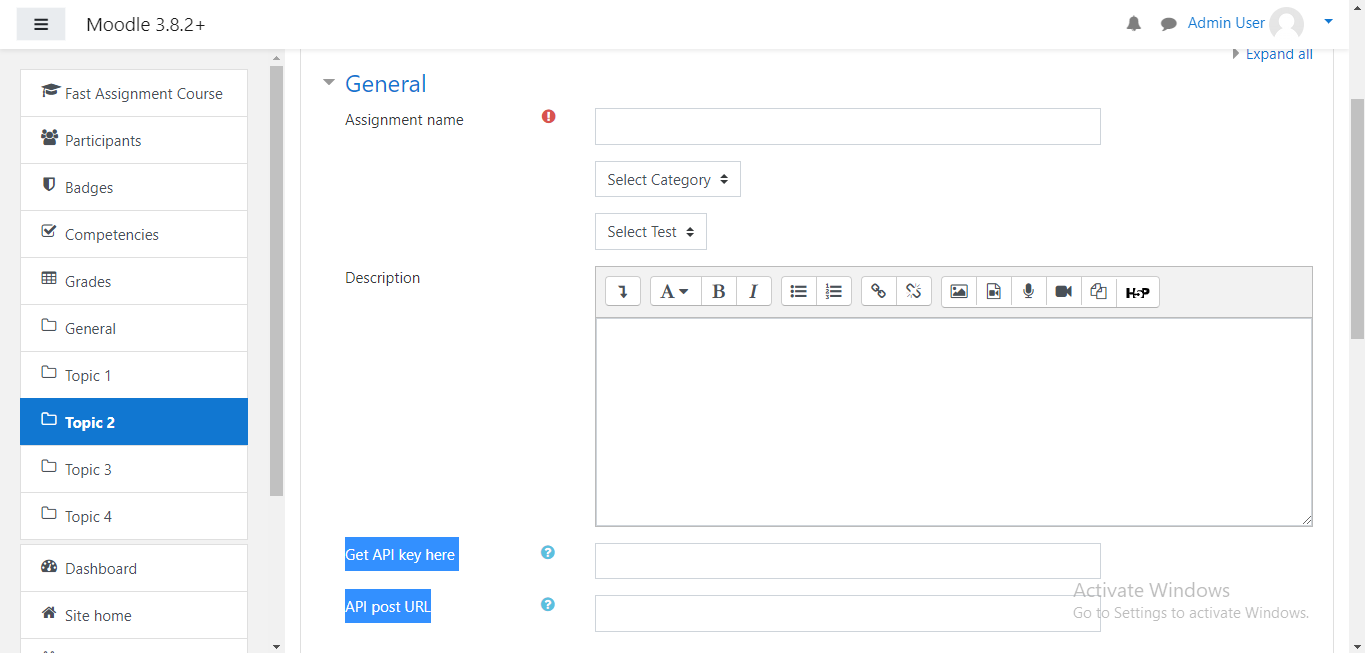
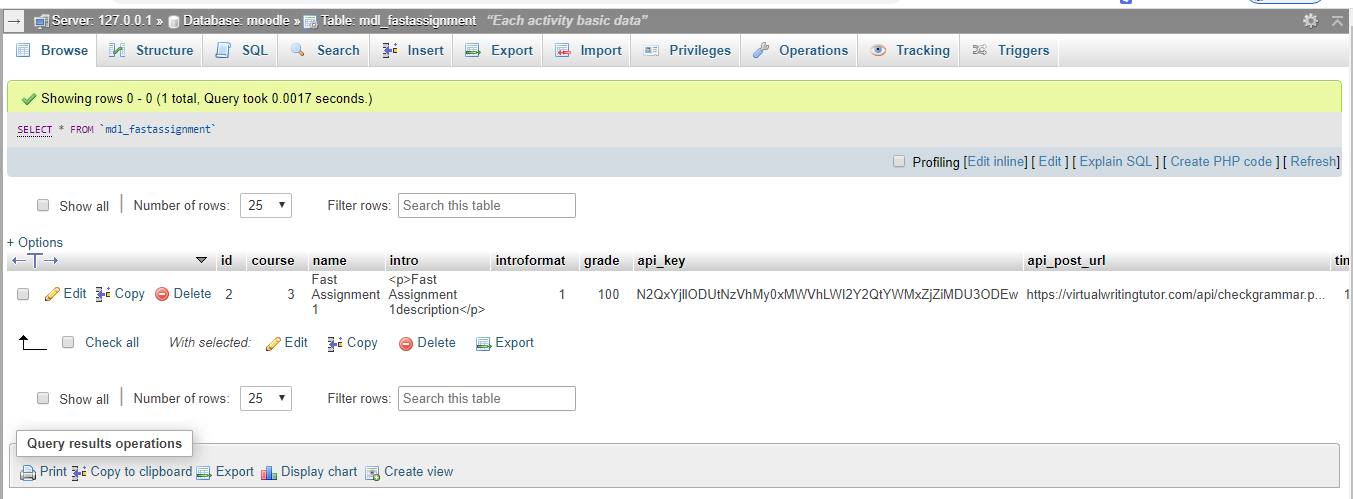
Moodle - Assignment Plugin Module

1. Create a basic assignment plugin Fast assignment - separate plugin
2. Basic Layout of fast assignment settings - created select category, select tests, get API key here, API post url. Admin form creation.

Added help info strings in the form



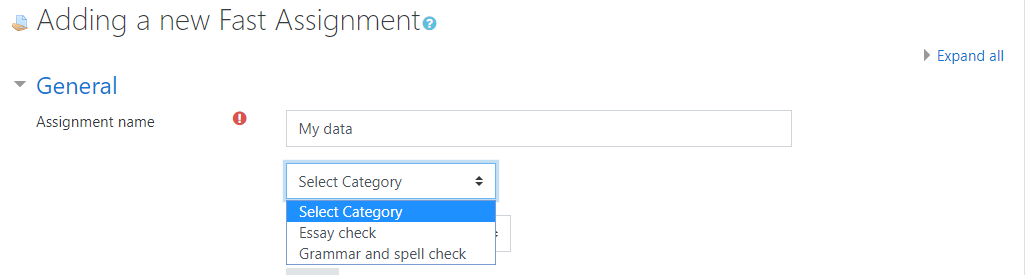
1. Database management and tables creation related with this form - storing encoded api\_key and api post url. (Not saving categories and tests)

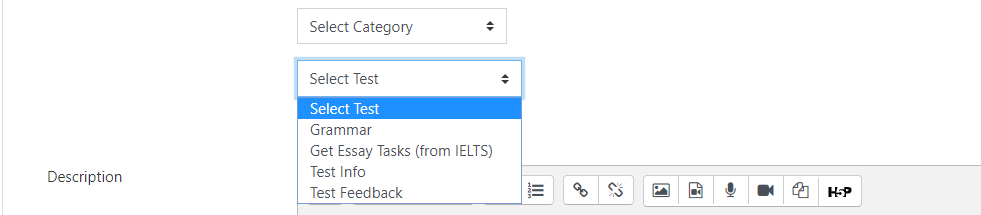


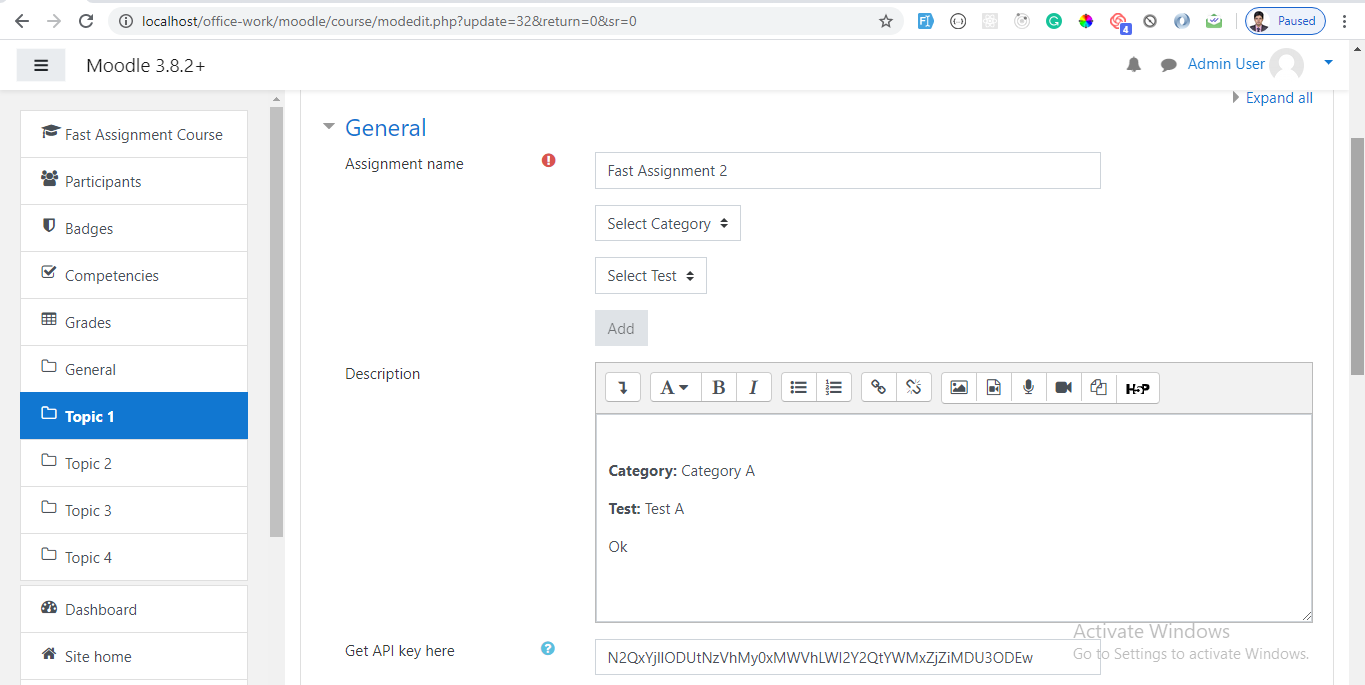
1. Data from moodle test dropdown to description Atto Editor. On clicking the add button data is imported to the Atto editor.

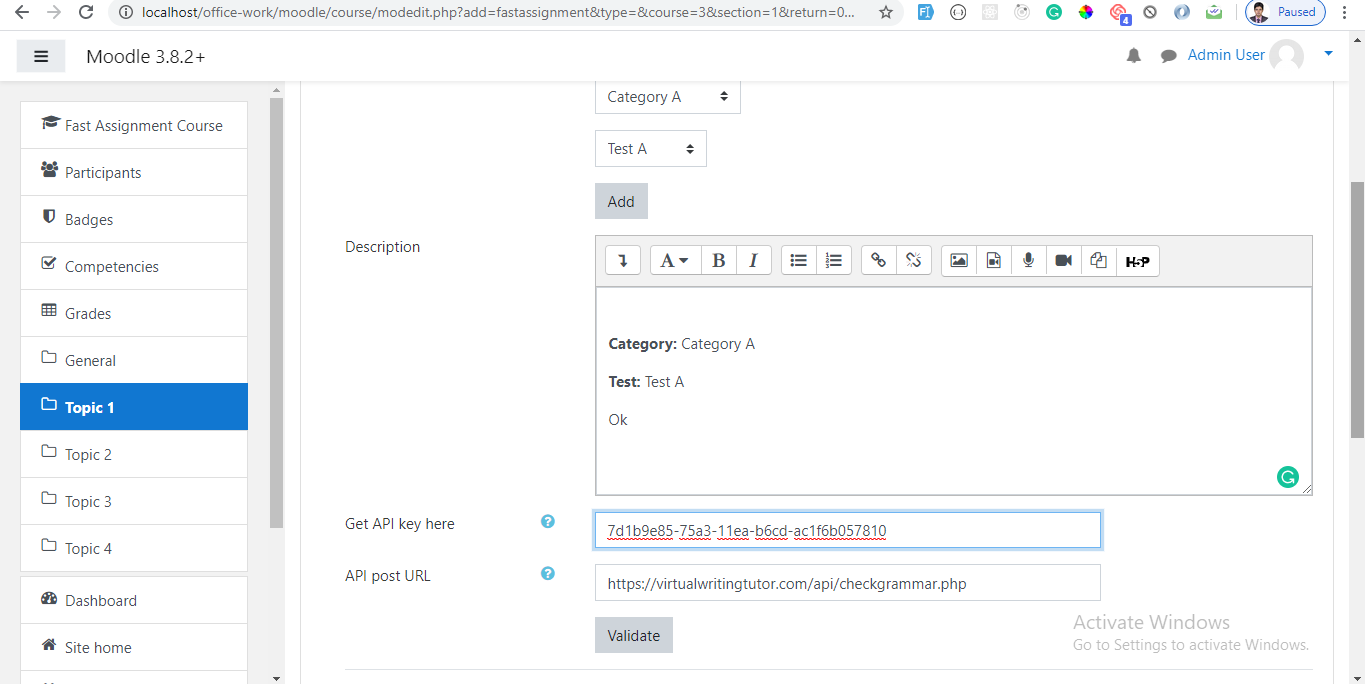
Categories: Grammar , Essay Check

Tests: Grammar. Get Essay Tasks, Test Info, Test Feedback



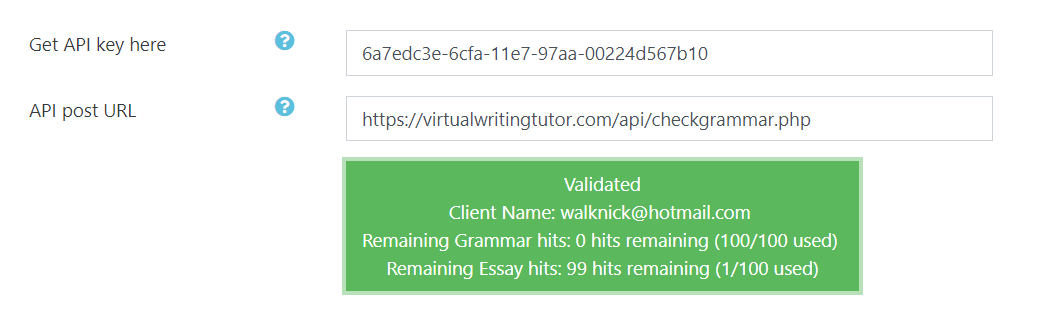


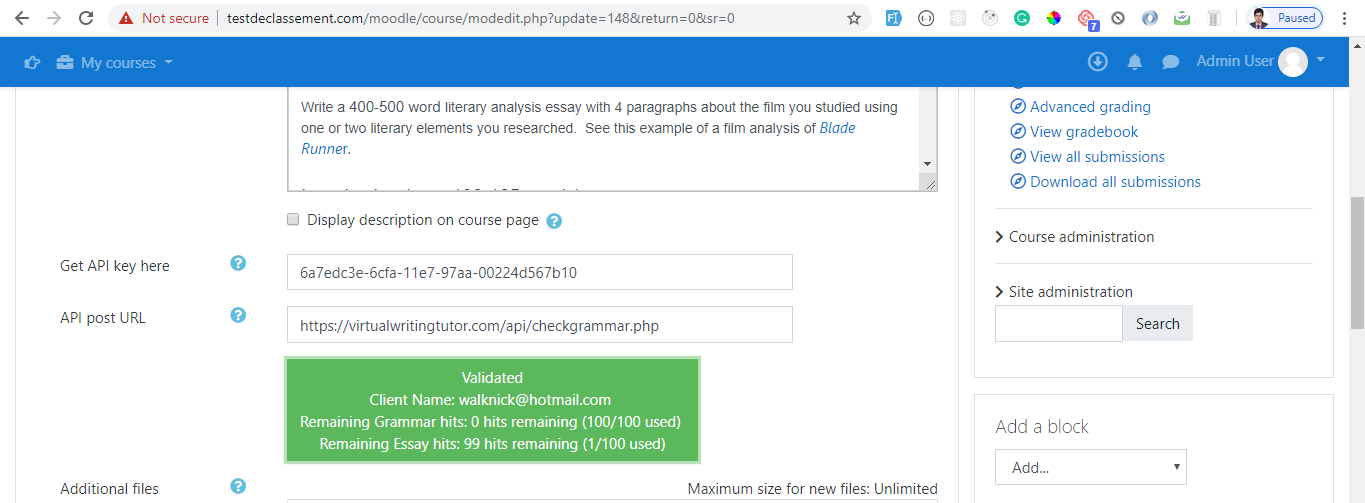


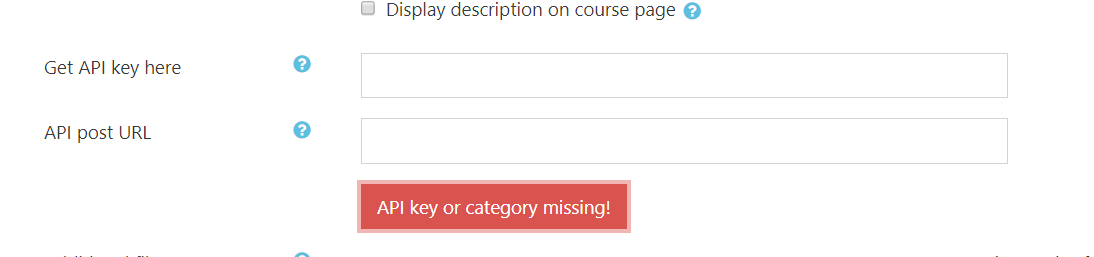


Video - <https://www.loom.com/share/80b9e251fab74270806d3e61d69aab1d>

1. Validate functionality - An API to check whether the teacher has number of hits available - check API for subscription plan

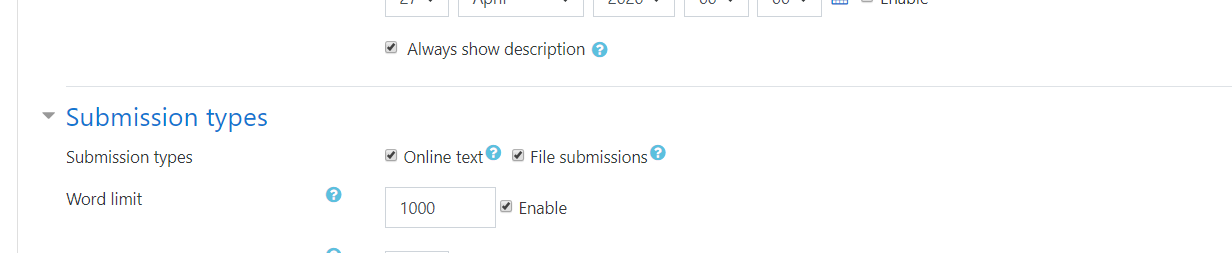






Feedback types

1. The submission type for enhanced assignment should probably be limited to online text of 1000 words.

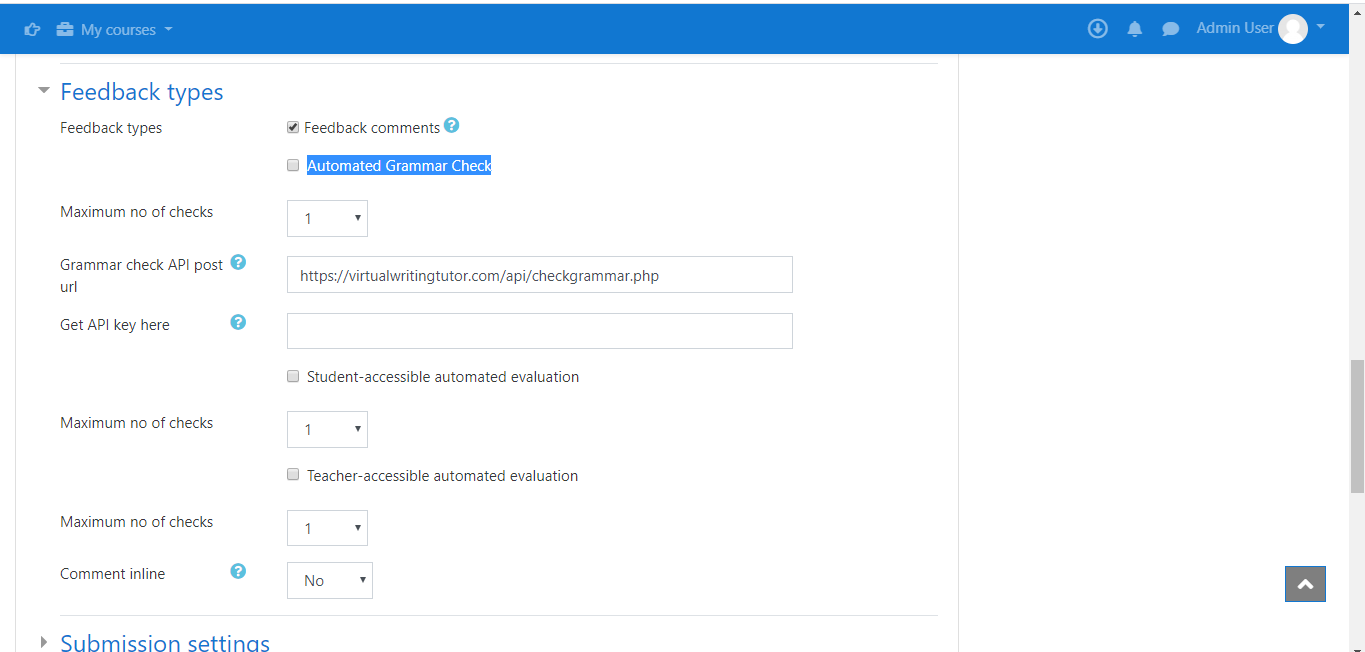


1. Automated grammar check,

Maximum no. of checks, API post url and API key

Student -accessible automated evaluation, Maximum number of student checks,

Teacher-accessible automated evaluation layout done



1. Database management for point 7

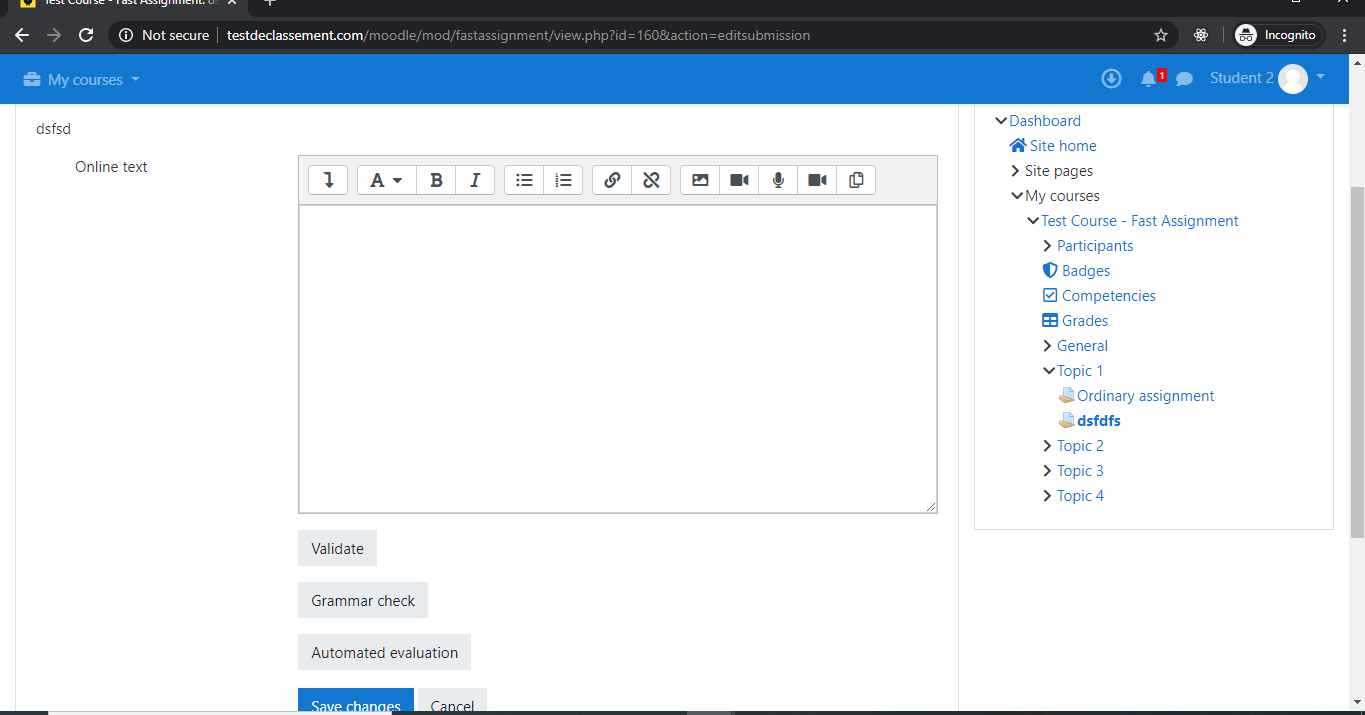
This is the portion where mform handle part is done (core code) - updated in live server

1. Grade - convert IELTS to percentage and vice-versa

Student view

**Validate, Grammar check, Automated evaluation, save changes(default)** and cancel(default) button is showing.

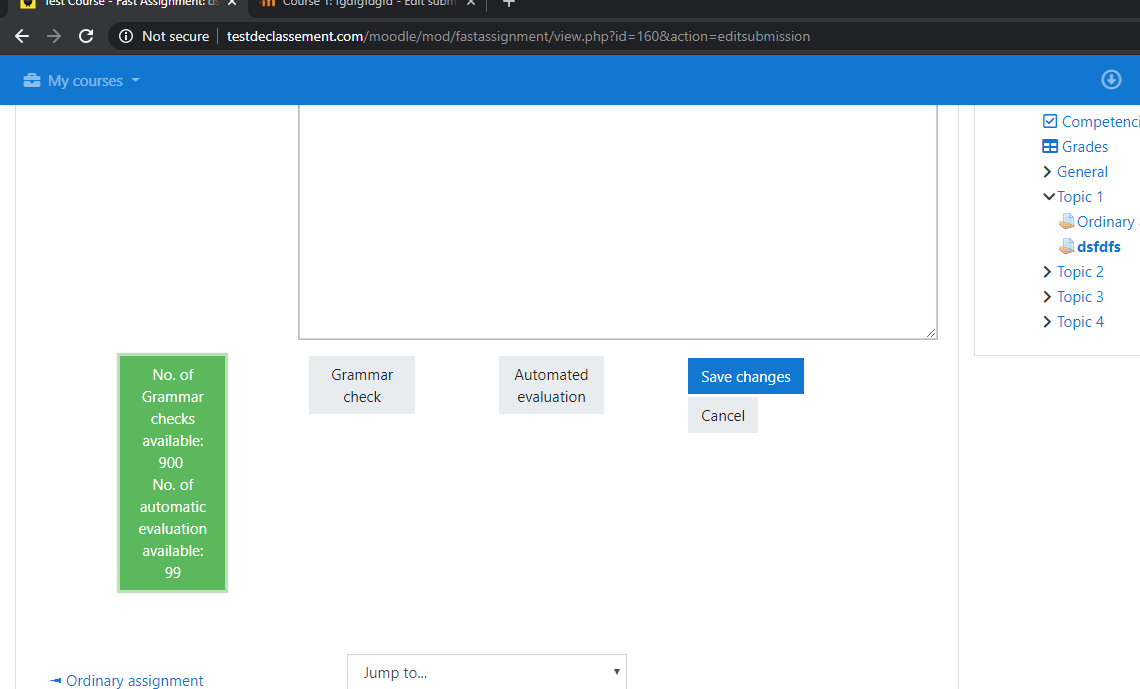
On clicking Validate sees list of api information like **grammar hits left and automated evaluation**

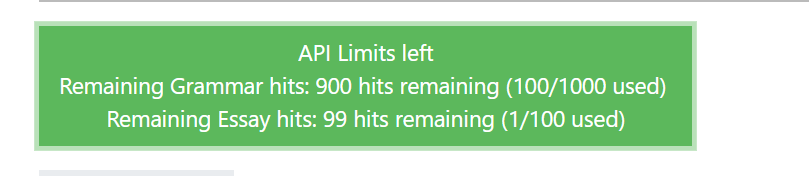


Check grammar functionality done. When student clicks Add submission button goes to the online editor. There he/she sees four buttons

1. Validate
2. Grammar check
3. Automated evaluation
4. Save and cancel (default)
5. Validate -

On clicking this shows the No. of grammar and automated api hits left.



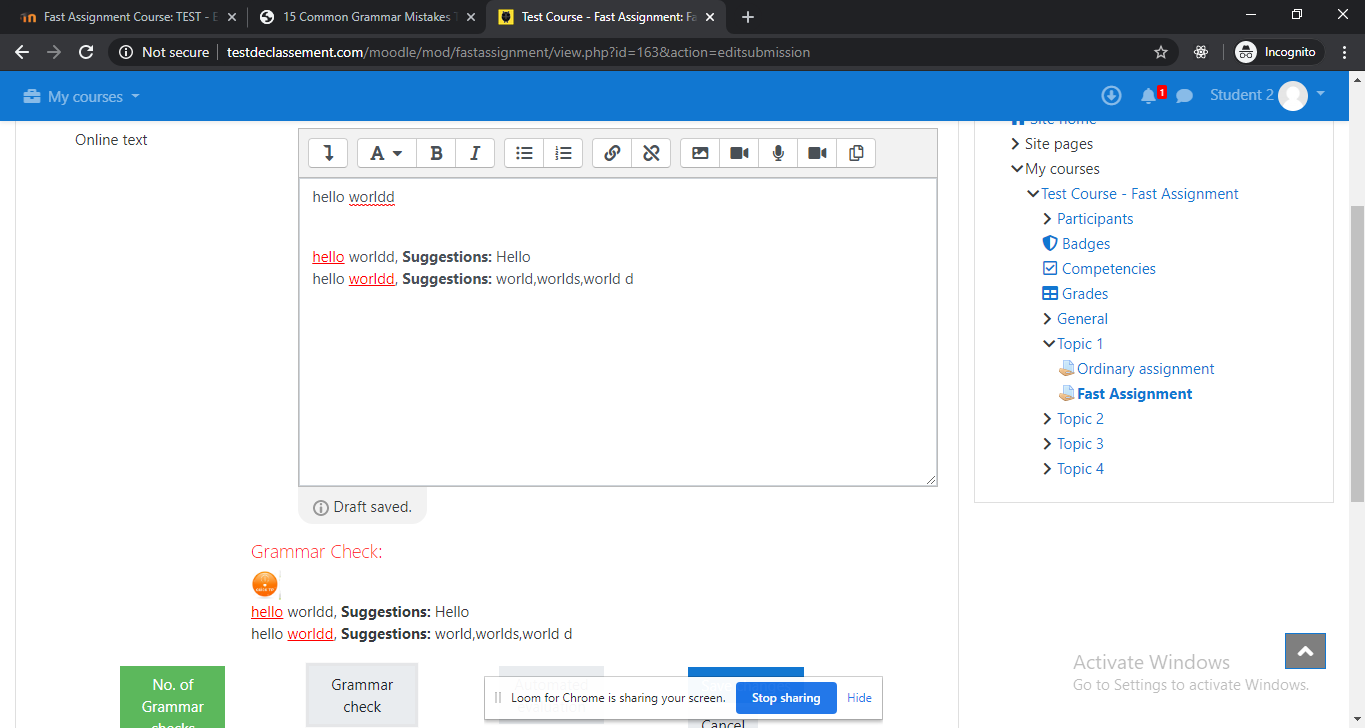


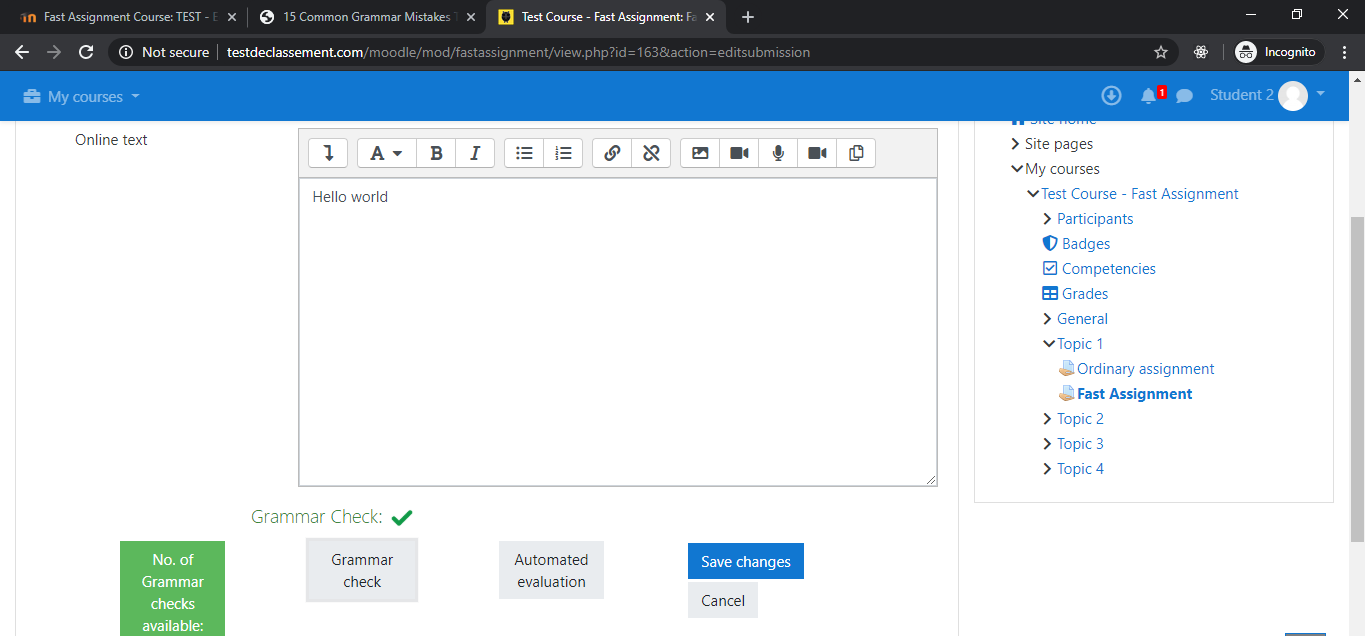
1. Grammar Check

When student writes in the editor and clicks Grammar check key, editor data is evaluated by Grammar check API

We get suggestions inside editor as well as above grammar check button. ( - submission\_form.php)

Video - [**https://www.loom.com/share/668c1f0e60064376940b9fcdea280dde**](https://www.loom.com/share/668c1f0e60064376940b9fcdea280dde)





1. **AUTOMATED EVALUATION**
2. Essay post API hit - successful. Getting response in code.

(With static test links )

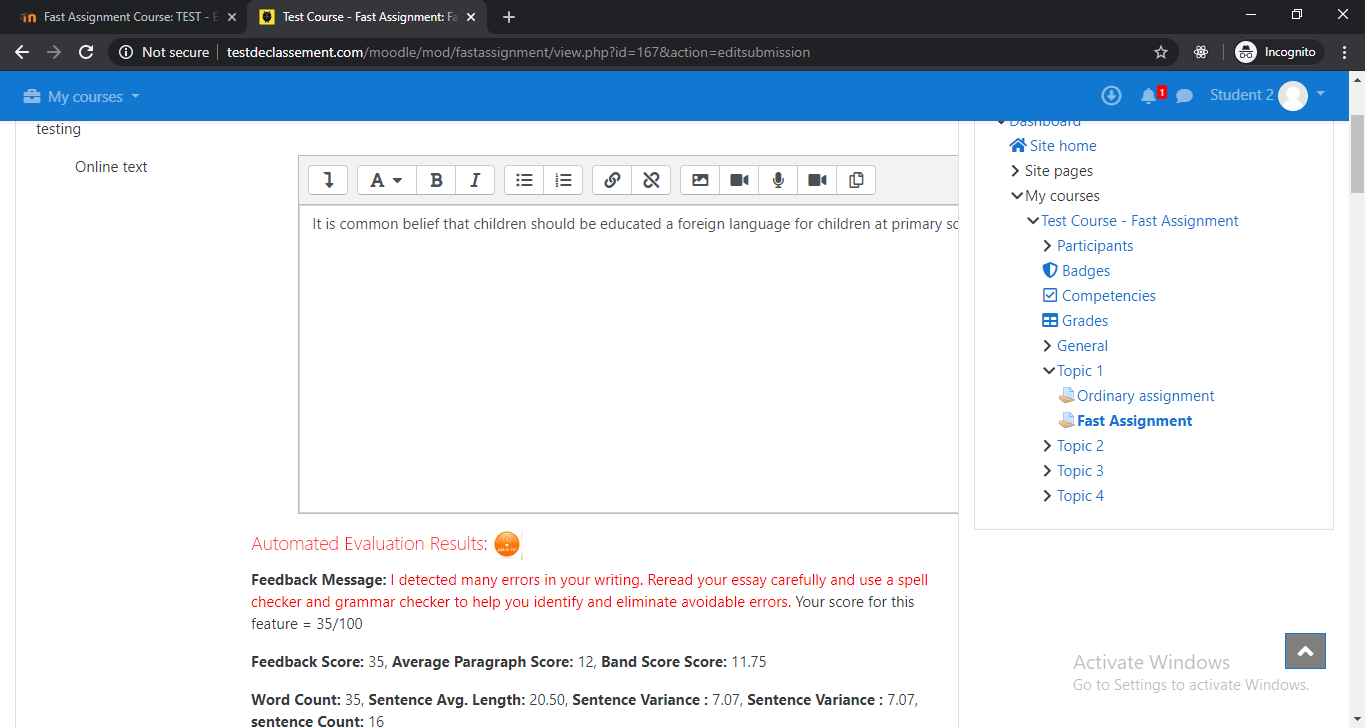
1. Generate Layouts for Automated Essay

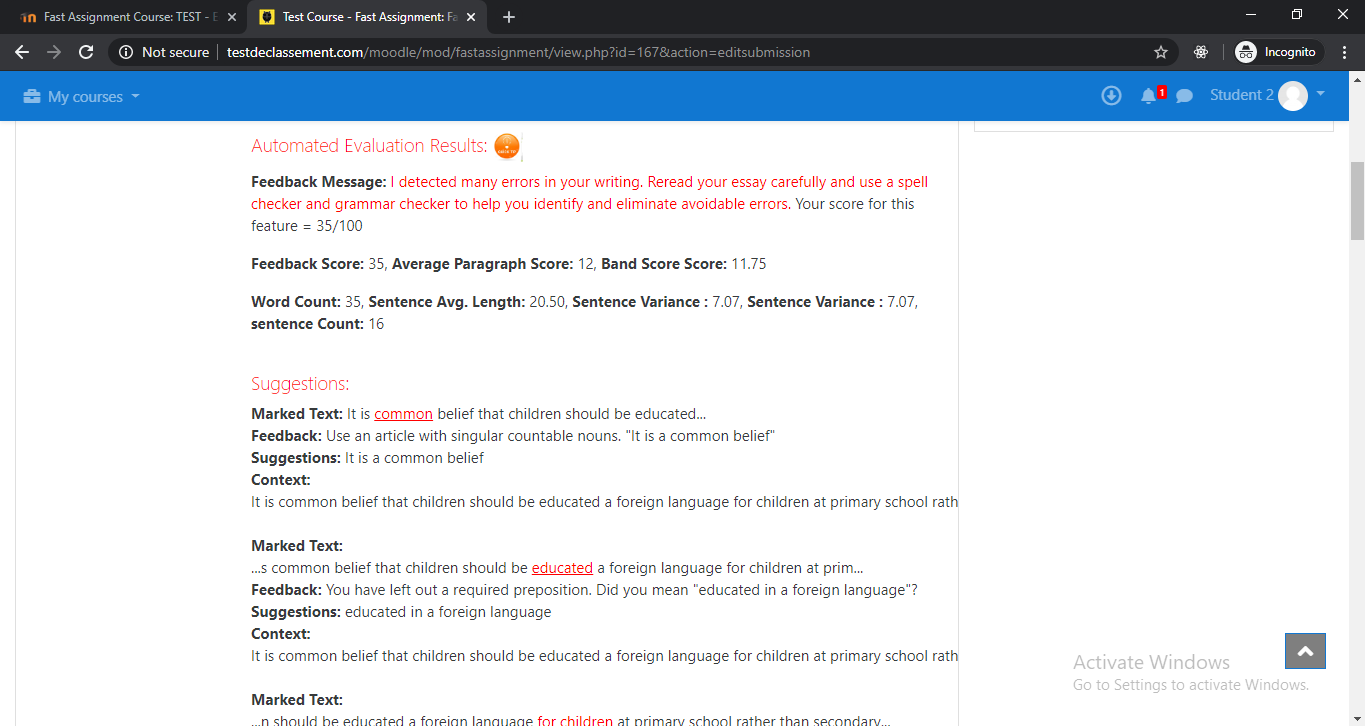
The overall design to create errors/suggestions. We need to know which data to use in suggestion response (getting very long success information - around 400 lines)

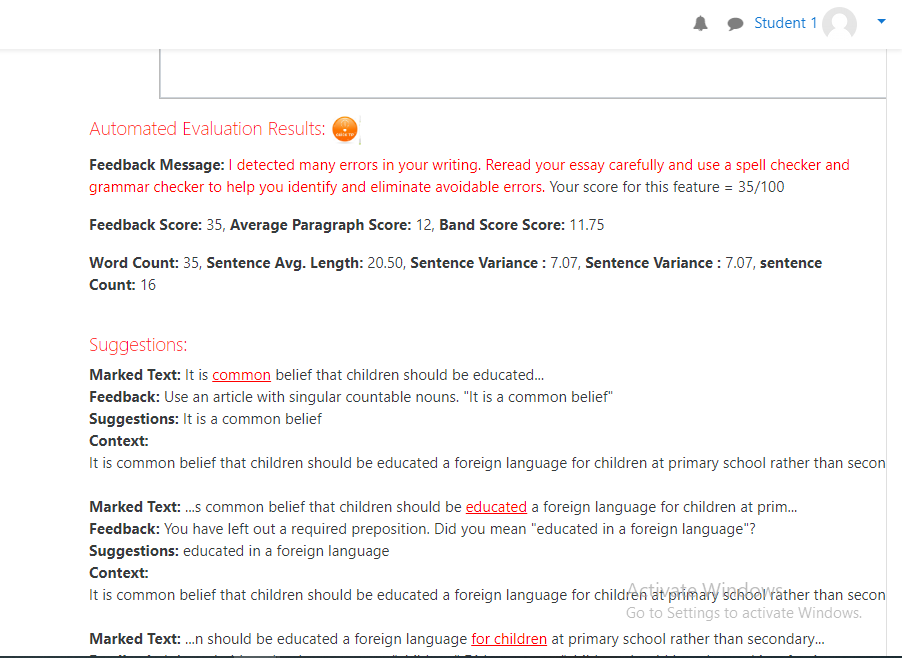
For the student view, API response for automated evaluation success result has

many keys and information

**Screenshots:**

****

****

****

**Video** showing all the **working 3 APIs of student view** - automated evaluation, grammar check and evaluate

[**https://www.loom.com/share/97d3ca01160a4f3089d9decd90d834fe**](https://www.loom.com/share/97d3ca01160a4f3089d9decd90d834fe)

1. In the database, **no test-links** are saved. Database management task of test-links need to be done

When admin/teacher creates fast assignment activity. Clients have given us in the shared doc to fetch the test and append data in the text editor and editor’s description value is saved in the database.

Now no test-links are saved. Test-links are used for POST API of automated evaluation

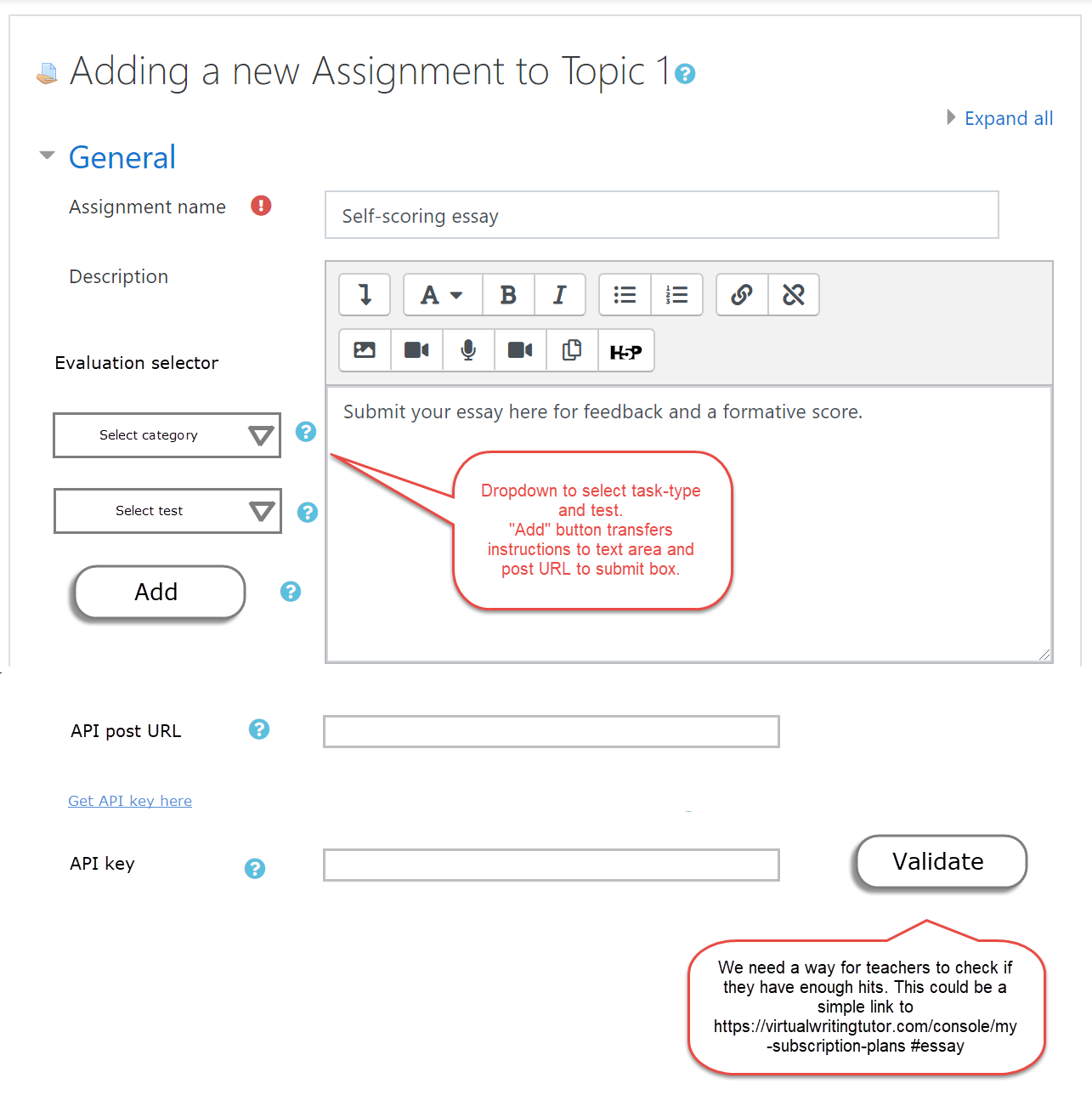
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Shared Requirements:

Moodle Assignment Plugin Module

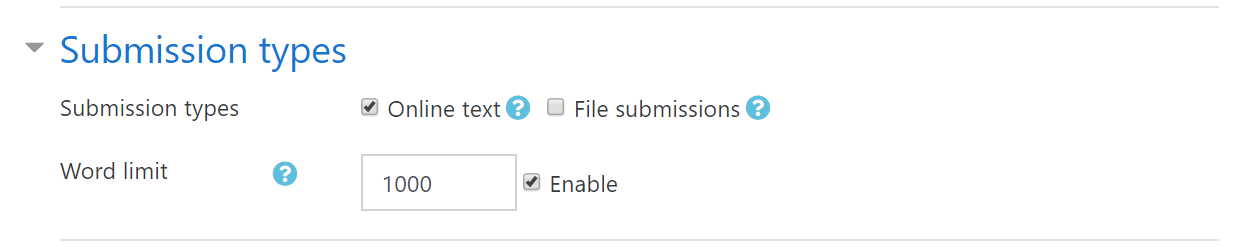
Assignment module name: Fast Assignment

Description: This modified Moodle Assignment allows students and teachers to get fast feedback on their writing through the integration of the Virtual Writing Tutor’s APIs. Users will need valid API keys to get fast feedback on writing submitted on the Fast Assignment Module. The Module will be available to download on the Moodle Plugin Repository for free.



If the Add Test Question button is used to fill the description field and the API post URL, we should make the checkbox on the Feedback Types section select automatically

The submission type for enhanced assignment should probably be limited to online text of 1000 words.



# Feedback types

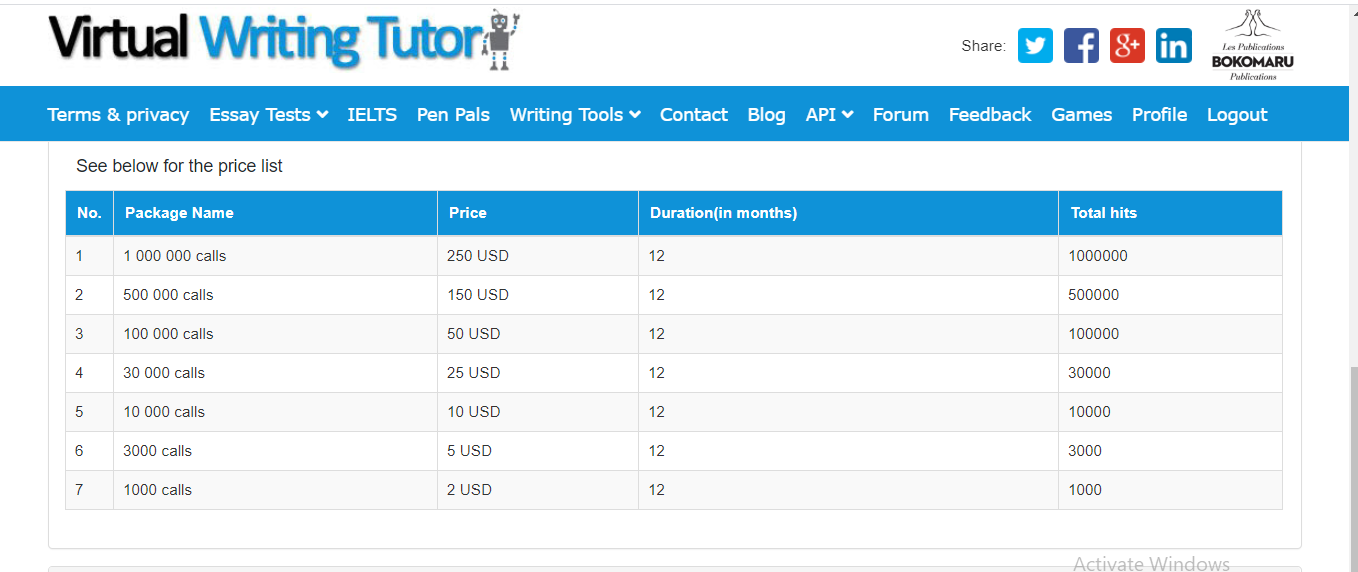
We could make it so that students can access the essay evaluation using the teacher’s API, but we would have to make it possible for the teacher to limit the number of essay checks since the teacher is paying for it.

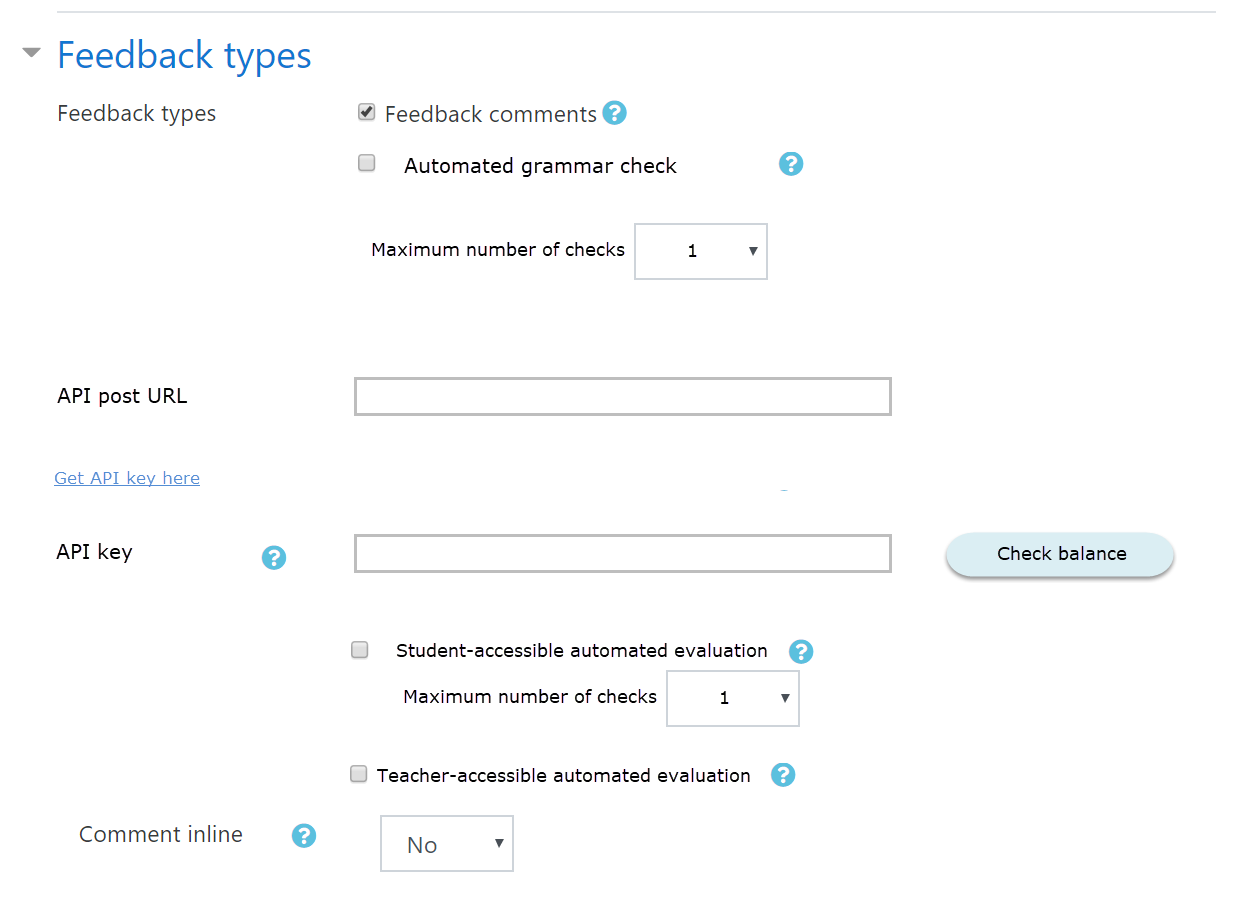
We could also make it so that the teacher can use just the grammar check API. I am sure there would be confusion, but we could add an explanation.

Alternatively, we could hide the essay check from the student and only let the teacher use the API to generate an evaluation.

In some cases, the teacher will want to select both choices.

Grammar checker has pricing plans:





The Essay check API URL should be added based on the choices from the dropdown fields

The grammar check API never changes. It is this:

When the teacher creates the assignment, they need to select the Essay Check API.

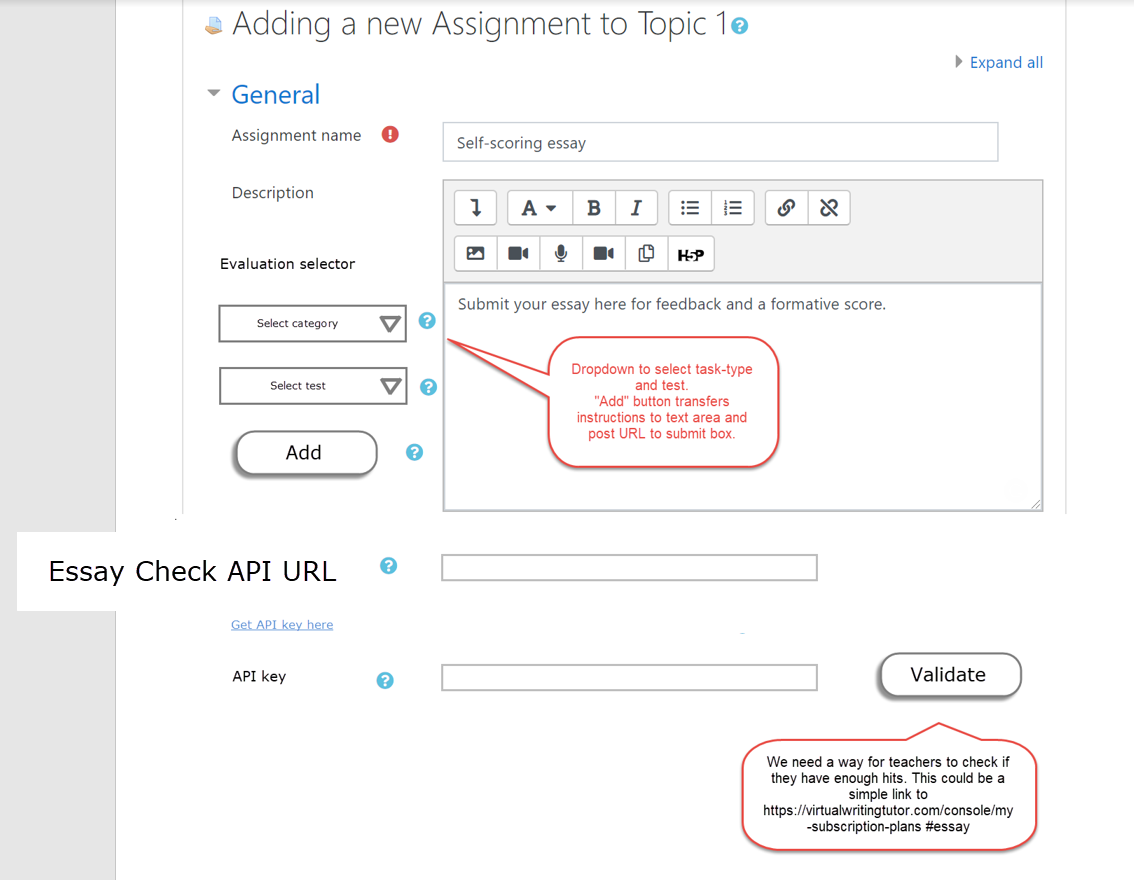
In the Feedback, they can choose to allow the student to send their text for just Grammar Check API before submission.

We need to add buttons to student text area so that they can access grammar check or Essay check based on the setting the teacher used when creating the assignment.

The idea is that the teacher uses the teacher's API key (kept secret) to create access to two forms of check.

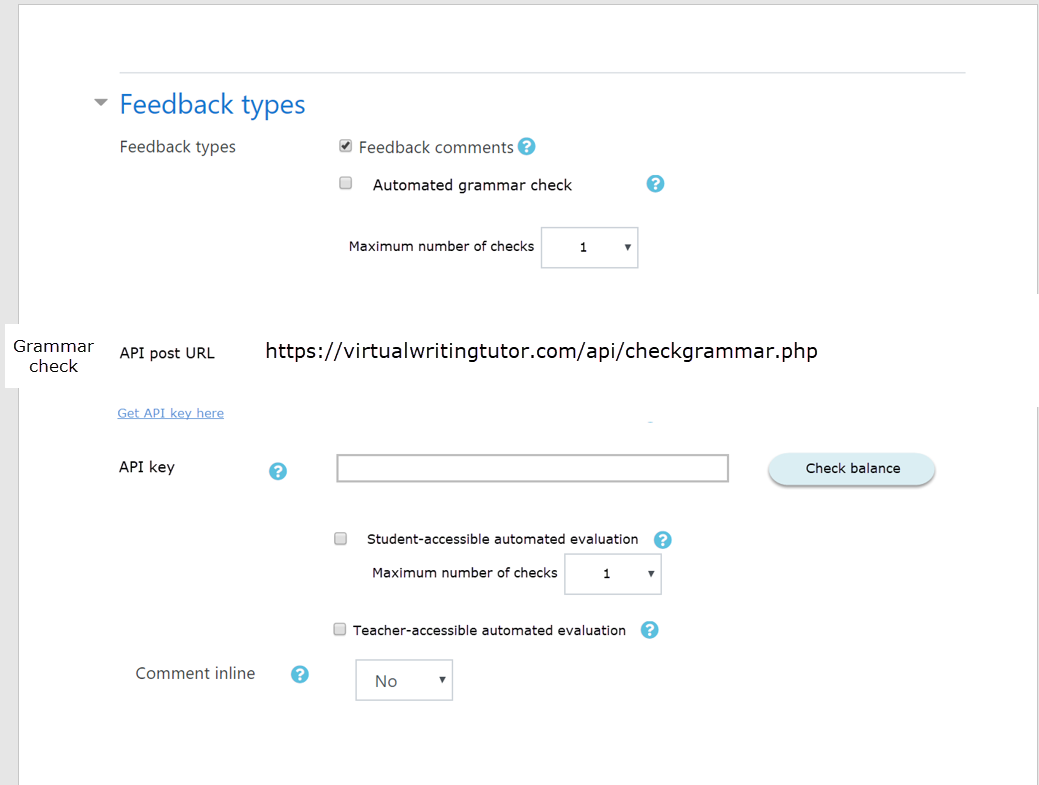
If the teacher selects it, the student will be able to get API feedback from the grammar check API and/or the essay check API.

If the teacher chooses, those buttons for grammar check and Essay check might not appear. Then only the teacher will have access during grading/scoring after submission.

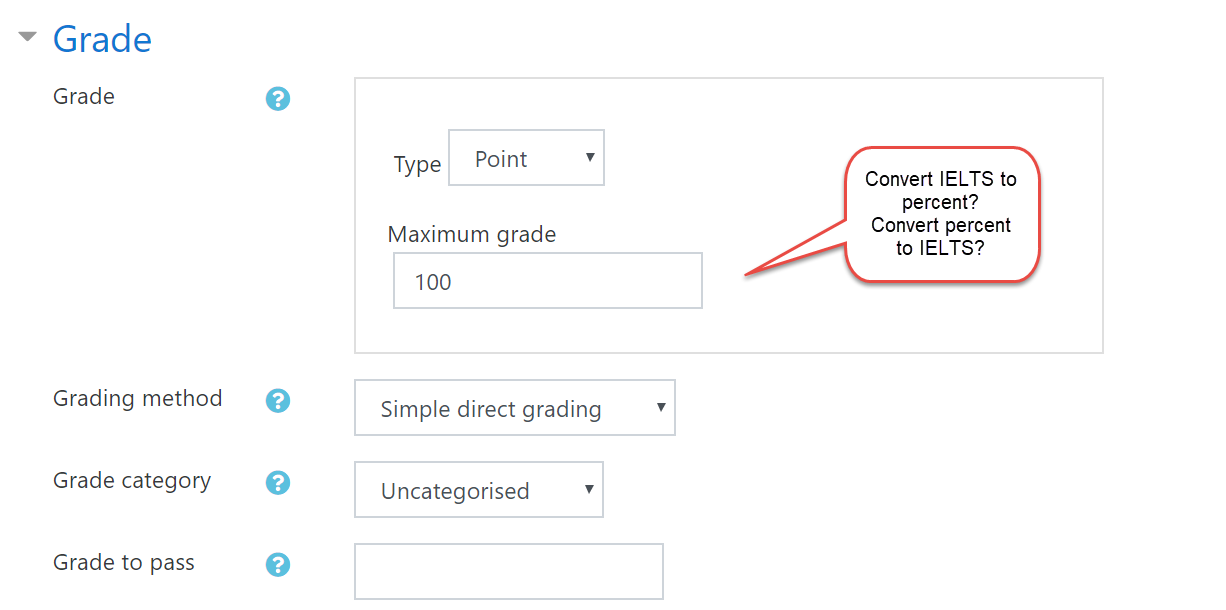


Select category is for Essay Check API. There are multiple post URLS because there are multiple essay types

There is only one Grammar check API post URL for any text sent



We might have to integrate an IELTS score conversion to a score on the Maximum grade

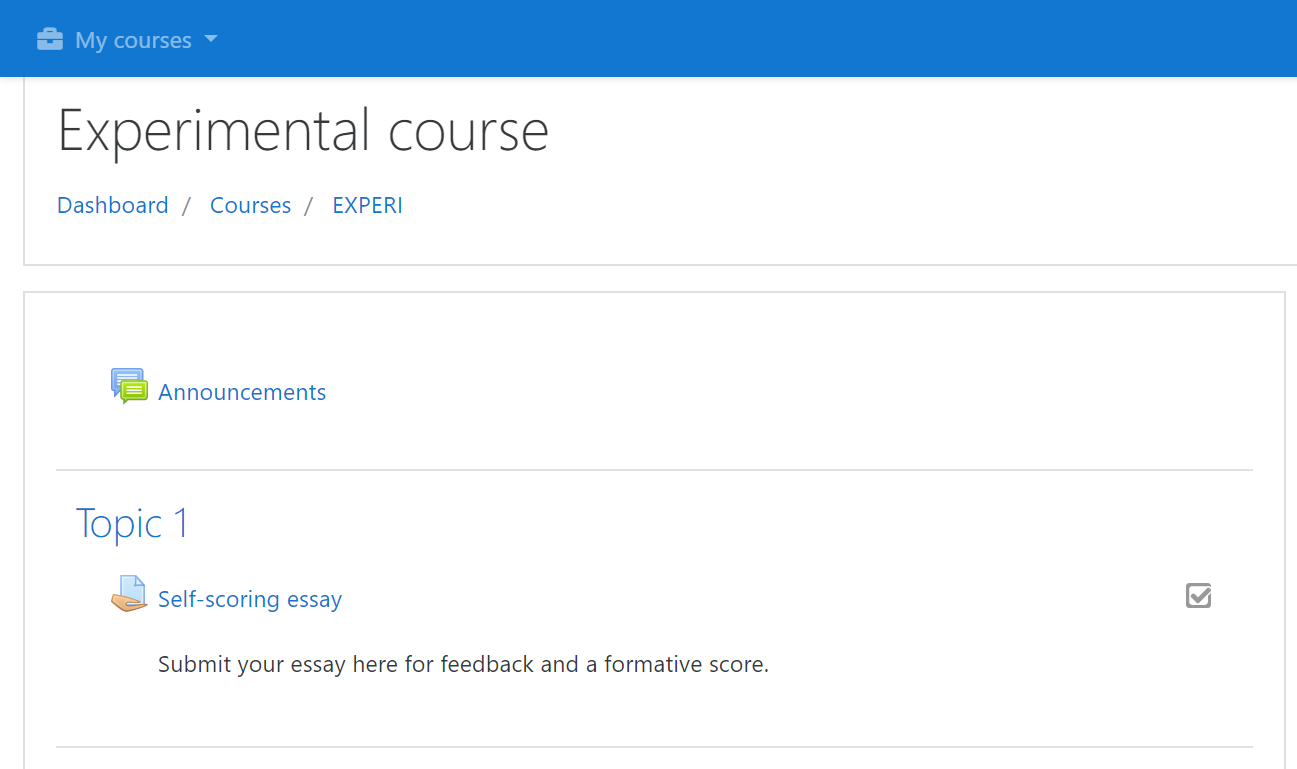


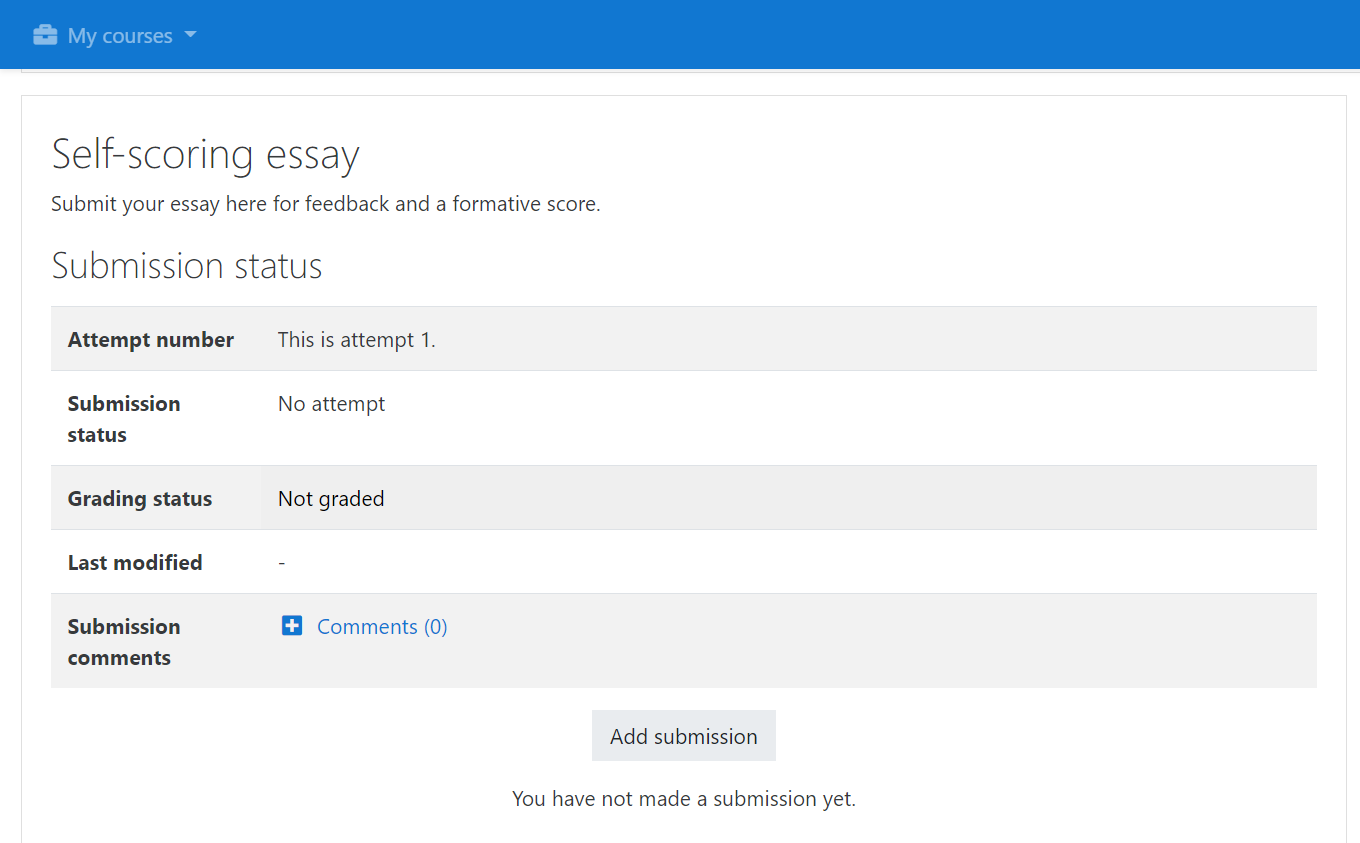
# Student view

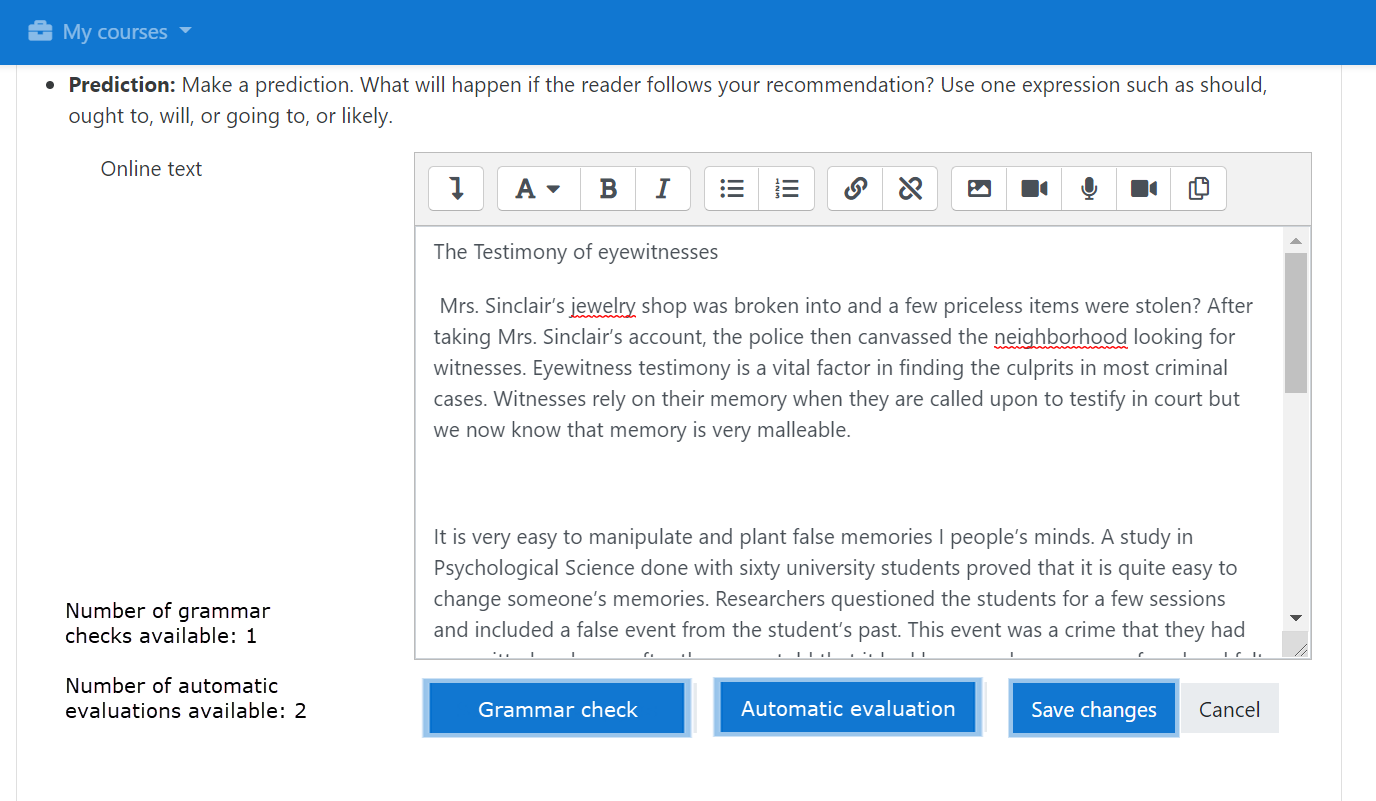
This is what the student sees

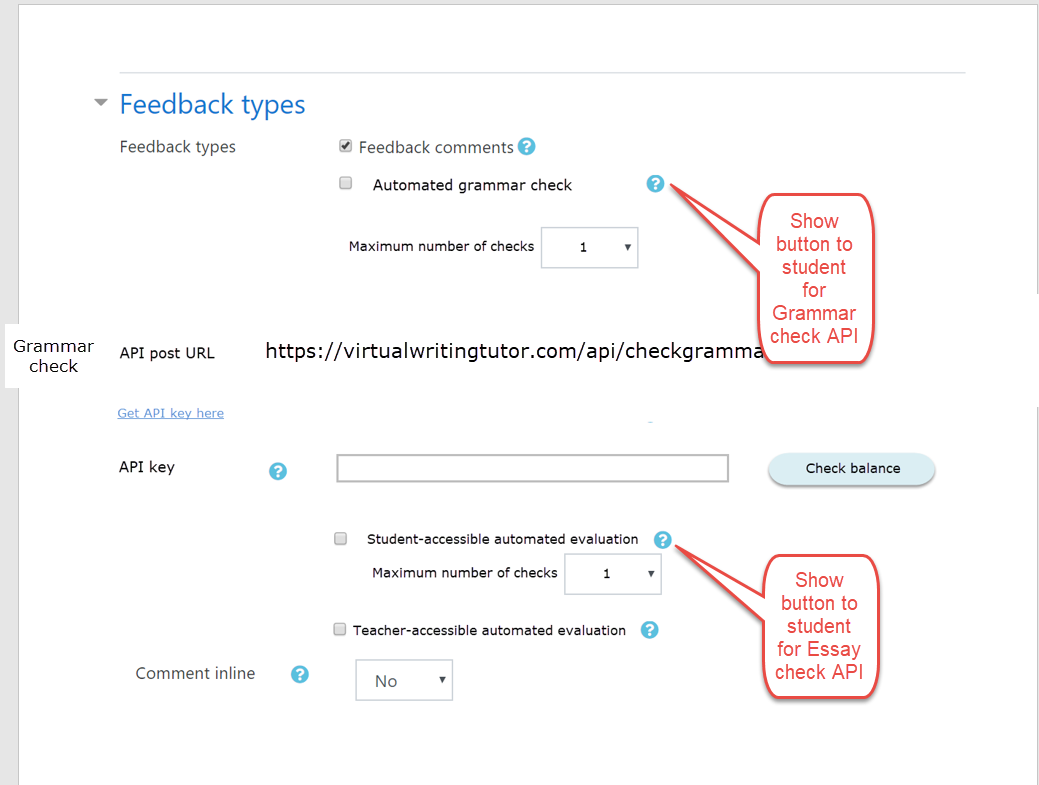
Username: student

Password: 5tud3nt-u53R

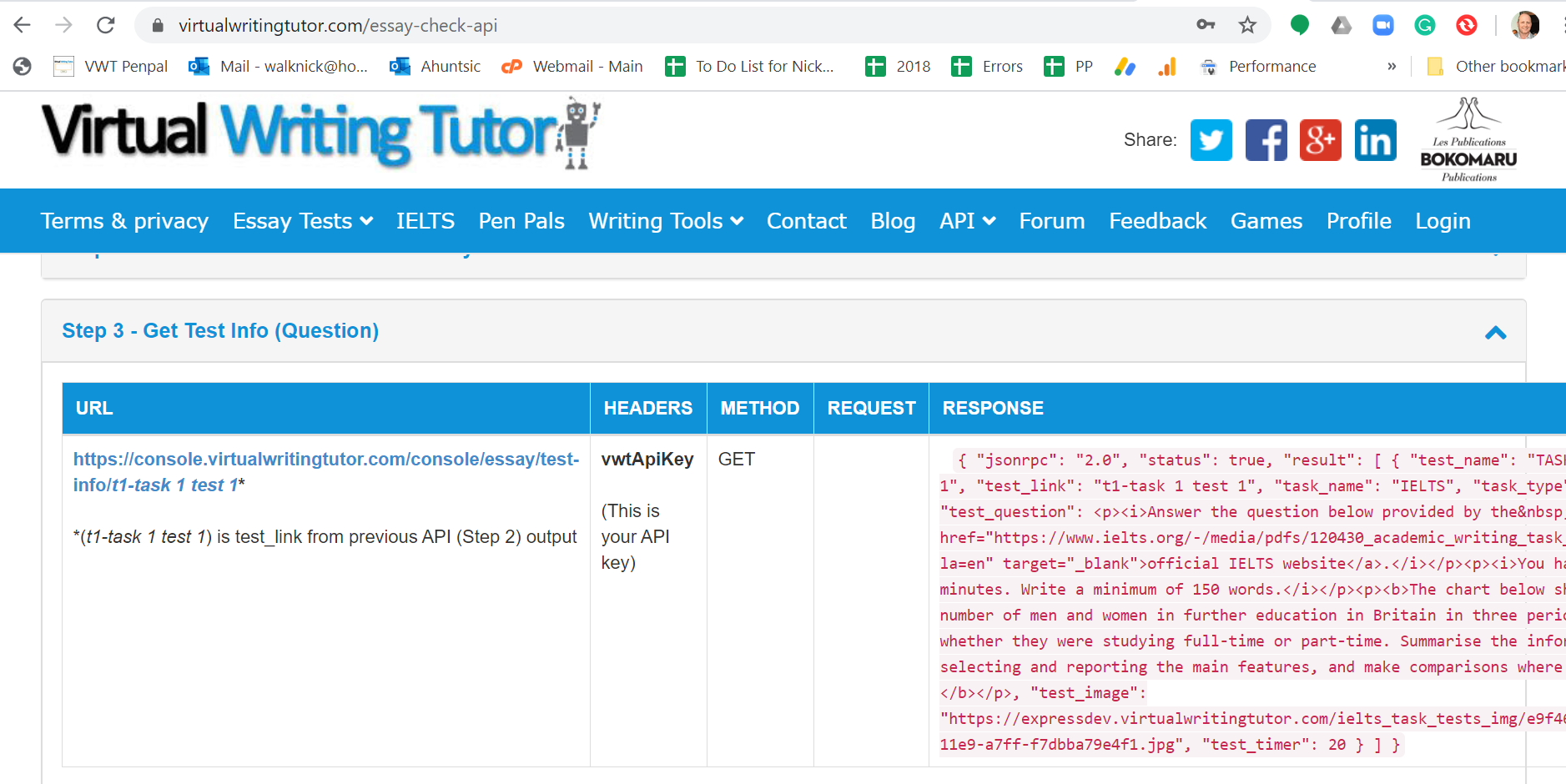








Here you can see that there is an API to get essay question instructions



so when they click on automated evalvation we will call the api and evalvate the essay question ? (This is for admin fastassignment setting results)

Yes. Clicking on "Automated Evaluation" button sends the text to Essay Check API using Post

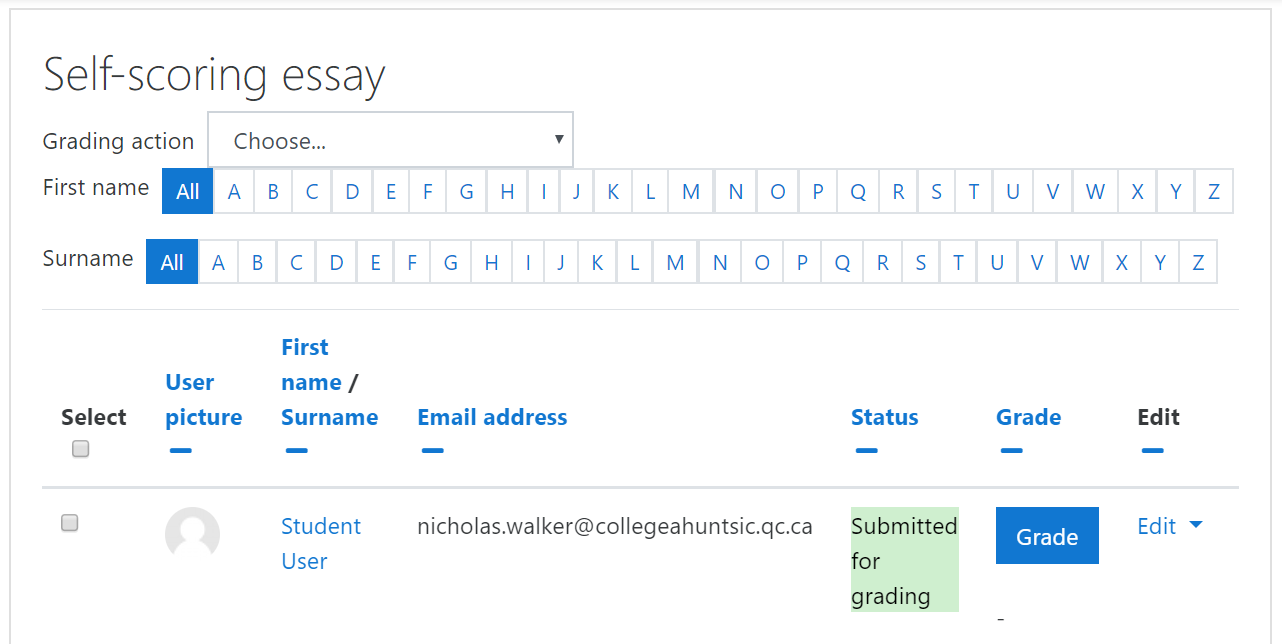
The web page should display the result under the submission text area.

So, I would like the Moodle plugin to fetch these instructions based on the dropdown selection. The instructions should automatically fill the Assignment creation text area.

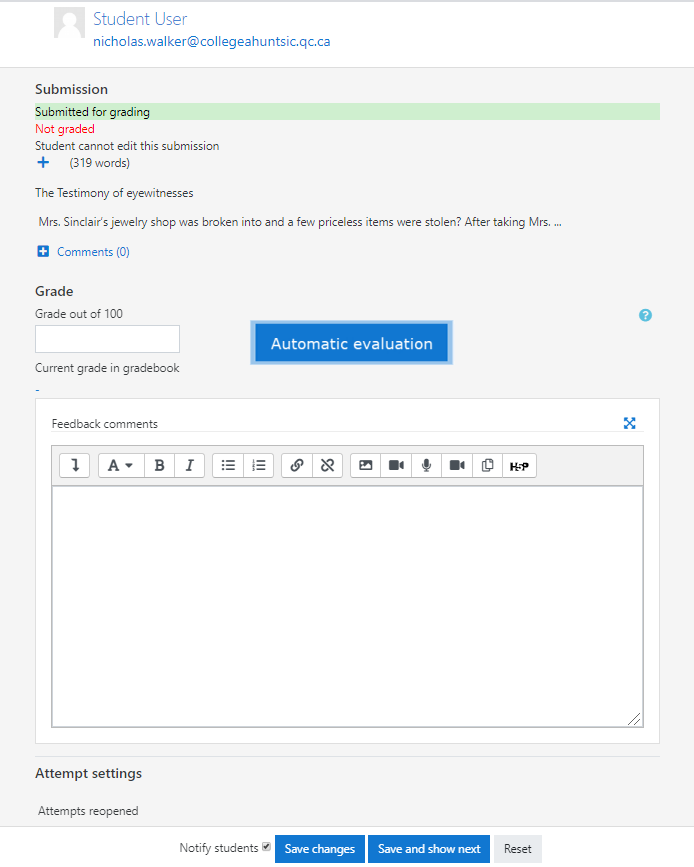
For the student view how shall I use the essay check. Can you provide me proper API which we will use for automated evaluation. We were using essay API to get description i.e. based on category, get tests and based on tests API get test info for editor description. So how we shall evaluate data like grammar?

# Teacher View

Teachers will need to score essays. They should use the automated evaluation and confirm or modify the score and comments.



After clicking “Grade”

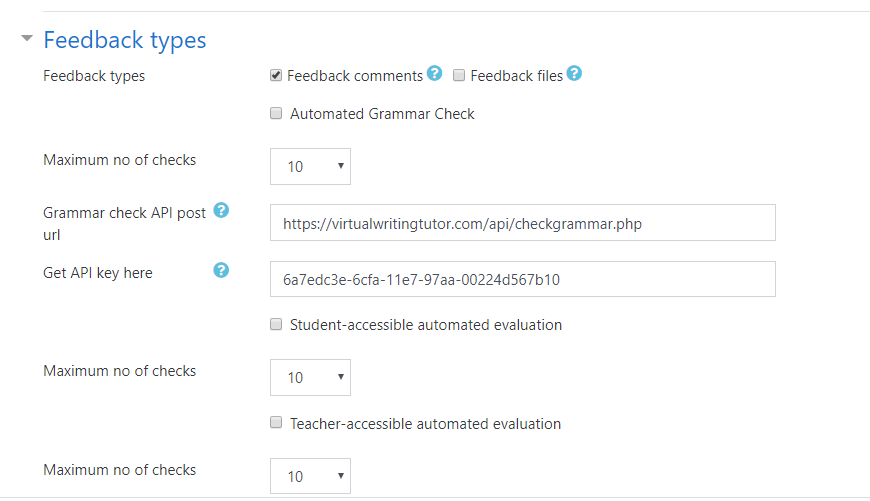


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Codes for Reference:

1. For feedback types extra configuration in form - Fast assignment settings

In the admin view edit setting of Fast assignment, this codes creates form under feedback types, header. Functionality Includes -



**Path - mod\fastassignment\feedback\comments\locallib.php**

public function get\_settings(MoodleQuickForm $mform) {

$OPTIONS\_MAXCHECKS = array(

'1' => '1',

'2' => '2',

'3' => '3',

'4' => '4',

'5' => '5',

'6' => '6',

'7' => '7',

'8' => '8',

'9' => '9',

'10' => '10',

'11' => '11',

'12' => '12',

'13' => '13',

'14' => '14',

'15' => '15',

'16' => '16',

'17' => '17',

'18' => '18',

'19' => '19',

'20' => '20'

);

// Automated grammar check // amir

$mform->addElement('checkbox', 'ratingtime', get\_string('automatedgrammarcheck', 'fastassignment'));

// Max no of checks

$select = $mform->addElement('select', 'maxnoofchecks', get\_string('maxnoofchecks', 'fastassignment'), $OPTIONS\_MAXCHECKS);

$default = $this->get\_config('commentinline');

if ($default === false) {

// Apply the admin default if we don't have a value yet.

$default = get\_config('fastassignmentfeedback\_comments', 'inline');

}

$mform->addElement('selectyesno',

'fastassignmentfeedback\_comments\_commentinline',

get\_string('commentinline', 'assignfeedback\_comments'));

$mform->addHelpButton('fastassignmentfeedback\_comments\_commentinline', 'commentinline', 'fastassignmentfeedback\_comments');

$mform->setDefault('fastassignmentfeedback\_comments\_commentinline', $default);

// Disable comment online if comment feedback plugin is disabled.

$mform->hideIf('fastassignmentfeedback\_comments\_commentinline', 'fastassignmentfeedback\_comments\_enabled', 'notchecked');

}

String files

Path - mod\fastassignment\lang\en\fastassignment.php

$string['automatedgrammarcheck'] = 'Automated Grammar Check';

$string['maxnoofchecks'] = 'Maximum no of checks';

1. File - fastassignment/locallib.php

Seach - public function load\_plugins($subtype, $type) {

// $subtype="assignsubmission";

// $names = core\_component::get\_plugin\_list($subtype); // default case

$basepath = dirname(\_\_FILE\_\_);

$names = [

"comments" => $CFG->dirroot . "/mod/fastassignment/". $type ."/comments",

"file" => $CFG->dirroot . "/mod/fastassignment/". $type ."/file",

"onlinetext" => $CFG->dirroot . "/mod/fastassignment/". $type ."/onlinetext"

];

1. File - fastassignment/submission\_form.php

This page discuss related with the student view form functionality - grammar check

$fastassignment->add\_submission\_form\_elements($mform, $data);

$mform->addElement('html', '<div class="validation"></div>');

$mform->addElement('html', '<div class="success"></div>');

$mform->addElement('button', 'validate\_api\_limits', get\_string('validateapibutton', 'fastassignment'));

$mform->addElement('button', 'grammarcheck', get\_string("grammarcheck", 'fastassignment'));

$mform->addElement('button', 'automatedevaluation', get\_string("automatedevaluation", 'fastassignment'));

$this->add\_action\_buttons(true, get\_string('savechanges', 'fastassignment'));

Path - course/modedit.php - course edit form (

Automated Grammar Check

Maximum no of checks. etc

)

// =============== FAST ASSIGNMENT: Managing Automated grammar check, student evaluation, etc: Start ==============

if($fromform->modulename === "fastassignment") {

$addition\_config = new stdClass();

$addition\_config->id = $fromform->instance;

if(isset($fromform->api\_key) && $fromform->api\_key !== "") {

$addition\_config->api\_key = $fromform->api\_key = $fromform->api\_key;

}

if(isset($fromform->api\_key\_feedback) && $fromform->api\_key\_feedback !== "") {

$addition\_config->api\_key\_feedback = $fromform->api\_key\_feedback = $fromform->api\_key;

}

if (!isset($fromform->automatedgrammarcheck)) {

$addition\_config->automatedgrammarcheck = $fromform->automatedgrammarcheck = 0;

}

if (!isset($fromform->studentaccessibleautoeval)) {

$addition\_config->studentaccessibleautoeval = $fromform->studentaccessibleautoeval = 0;

}

if (!isset($fromform->teacheraccessibleautoeval)) {

$addition\_config->teacheraccessibleautoeval = $fromform->teacheraccessibleautoeval = 0;

}

$addition\_config->api\_post\_url = $fromform->api\_post\_url;

$addition\_config->apiposturl\_feedback = $fromform->apiposturl\_feedback;

$addition\_config->maxnoofchecks = $fromform->maxnoofchecks;

$addition\_config->maxnoofchecksstudent = $fromform->maxnoofchecksstudent;

$addition\_config->maxnoofchecksteacher = $fromform->maxnoofchecksteacher;

$table = "fastassignment";

if($DB->update\_record($table,$addition\_config)){

// echo "db updated success";

} else {

console.error("Error writing the fastassignment database");

}

}

// =============== FAST ASSIGNMENT: Managing Automated grammar check, student evaluation, etc: End ==============