Moodle Essay PDF Annotator: A Standalone Plugin

B.Tech Project presentation

IIT Palakkad

15 May 2024

Nideesh N 112001028
VM Sreeram 112001051

Mentor
Dr. Jasine Babu

Background

- Moodle is one of the most widely used open source LMS
- Modular design. Extend functionality using plugins, without modifying core
- PDF annotation for grading essays in quiz is a highly demanded for functionality in Moodle (MDL-65872[1])
- Two past attempts[2,3] – involved modifications to core code

Objectives

1. Develop a standalone Moodle plugin for PDF annotation for grading essays, reusing existing code
2. Address security concerns from Moodle developers[1]

Design Changes and Improvements

- Detaching the code from Moodle core – design decision to add a new question type `essayannotate`
- Security aspects and role based user access control
- Backup and restore, import/export of questions
- Adherence to coding style and conventions mentioned in the checklist[4,5]
- Automated testing

Security Aspects and Role Based User Access Control

- Login and capability checks
- The Annotate button is shown only to users with capability to grade, only in grading page
- Corrected documents visible to graders always, but to students only after grading is completed
- User data sanitization – POST parameters
- Shell commands – reduction and sanitization
- Disabling browser access to internal pages

Backup and Restore

- Used by admins/teachers to port their courses across Moodle installations
- The backup plan consists of multiple tasks and steps – organized and executed in a predefined order

```
activity → quiz → quiz attempts → quiz attempt → question usage →
question attempts → question attempt → steps → step
```

- While backing up a quiz, every question attempt gets backed up along with corresponding file attachments
- The essential functionality of backup/restore of annotated files was not addressed in the previous works

Backup and Restore – Relevant Data Structures

- Question attempt – student's response to a particular question
- Question attempt step – interactions and actions taken by both students and teachers during the life cycle of a question attempt
- Question attempt step data – detailed information associated with each step
- Question usage by activity – collection of all question attempts that make up a quiz attempt
- `mdl_files` – table in Moodle that stores information about files uploaded. Item id is an attribute in this table

Backup and Restore – Logic

- Similar logic used for file uploads in quiz is implemented for annotated files
- Question attempt step – responsible for marking the submitted files for backup, item id of the file in `mdl_files` table is set as the id of this step
- After each annotation, an annotation step is added. For each annotated file, item id is set as the first annotation step id
- `question_usage_by_activity::process_action()` on `$quba` is used to create the step

Import and Export of Questions in XML Format

- Useful for sharing questions among different quizzes or across courses
- Implemented `import_from_xml()` and `export_to_xml()` in `questiontype.php`

Multilingual Support (Internationalization)

- Use of string API
- `get_string()` function with component name and identifier

Third Party Libraries

- Fabric JS, FPDF/FPDI, PDF JS, and PDF JS Annotations
- Details declared in `thirdpartylibs.xml`
- Removed libraries that are already shipped in core

Adherence to Coding Conventions and Styles

- Code checker – A plugin to check adherence to coding guidelines for PHP
- Modularization of JavaScript
- Grunt – A tool that perform tasks such as minification, compilation, and linting of JavaScript files
- Coding style adherence for Gherkin, CSS, and Mustache files as suggested by Moodle continuous integration tools

Modularization of JavaScript: Vanilla JS to AMD JS

- All library files should be located locally for offline support and for performance reasons. Past works used `<script>` tags; libraries in CDN
- Moodle allows ESM/AMD format for modularization. But one of the third party libraries, FabricJS, does not support ESM
- AMD format
  - Wrap the definitions for modularity
    `define(module_id,[dependencies],function definition)`
  - Suggested way in Moodle to invoke JS calls
    `$PAGE->requires->js_call_amd()`
  - Logic for the JavaScript call is present in page footer
- All JS is served from `amd/build` directory after minification of source files located in `amd/src` directory

Templates to render HTML

- Moodle requires using Mustache templates instead of plain HTML which supports dynamic content rendering
- Created a template for the annotator UI
- Code changes for rendering Mustache from PHP

```
$output->header()
$output->render_from_template()
$output->footer()    // for js_call_amd
```

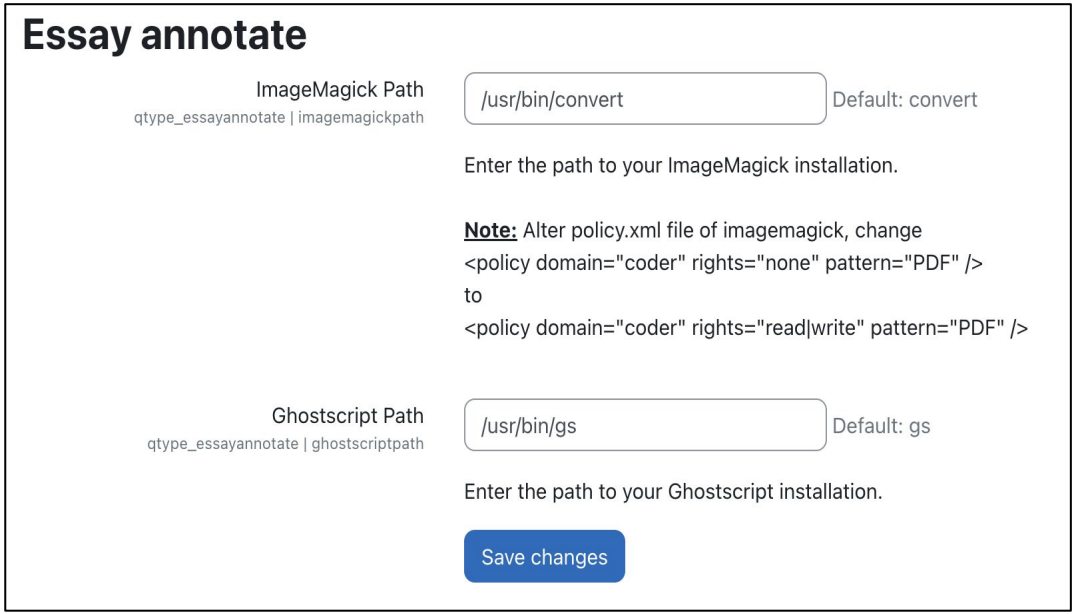
- UI simplifications

CSS organisation

- All CSS styles in `styles.css`
- CSS from all plugins – concatenated and served as one big resource
- Namespacing of CSS selectors. For example, `.messagebox` will become `.qtype_essayannotate_messagebox`

Settings Page

- Two dependencies – ImageMagick and Ghostscript
- Shell is used to execute the commands
- Admins can set paths to the executables



The settings page of the plugin

Annotation comment

Response history

Step	Time	Action	State	Marks
1	10/05/24, 10:40:55	Started	Not yet answered	
2	10/05/24, 10:41:20	Saved: Attachments: quiz ans q1.pdf (3.1 MB)	Answer saved	
3	10/05/24, 10:41:53	Attempt finished	Complete	
4	10/05/24, 10:44:37	Commented: Annotated file: quiz ans q1.pdf Nideesh N	Complete	
5	**10/05/24, 10:44:37**	**Commented: Teacher has started grading** **Nideesh N**	**Complete**	

Response history table after first annotation by teacher

Testing the Plugin

Automated tests

- Automated acceptance tests using Behat[6] – a framework for behavior driven development (BDD)
- Functionalities can be specified as a human-readable list of steps
- Scenarios covered
  - Student uploads a PDF file and teacher is able to annotate
  - Student uploads a PNG file and teacher is able to annotate
  - Student can see the annotated file only after the question is graded
  - When the quiz is backed up, the annotated files also gets backed up
  - Questions can be imported and exported to XML format

Testing the Plugin

Manual testing

- File size near limit
- Incorrect paths to dependencies
- Lack of write permission to EssayPDF directory
- PDF with versions higher than 1.4
- Unsupported file types
- Working in Moodle versions 4.0 to 4.4 and PHP version 7.4 and 8.1
- Cross-db compatibility with PostgreSQL and MySQL

Bug Fixes

Description	Fix implemented
Concurrent annotation of question attempts	Append attempt id, slot and user id to filename of dummy file
Attachments with the same file name for different questions	Add usage id and slot to fileinfo array
Missing essayPDF directory	Create the directory with proper permissions, if it doesn't exist
File extension/MIME mismatch	Check consistency between MIME type and extension
Filenames containing whitespace	Enclose file names within quotes
Images getting cropped	Shrink/expand files to A4 during `convert`

Conclusion

- Completed the development of a standalone quiz annotation plugin
- Security concerns were addressed
- Adhered to Moodle plugin development guidelines and coding style
- Support for Moodle versions 4.0 to 4.4 with PHP 7.4 and 8.1
- User documentation for plugin in Wiki page of repo
- Submitted to Moodle, approval awaited
  https://moodle.org/plugins/qtype_essayannotate
  https://github.com/vmsreeram/moodle-qtype_essayannotate

Possible Future Work

- Capability to edit annotations from past sessions
- Assignment annotation – fixing deprecated module

References

[1] Moodle tracker. Apply pdf annotation for grading essay or written response quiz questions. URL https://tracker.moodle.org/browse/MDL-65872. Accessed 14-May-2024.

[2] Parvathy S. Kumar and Asha Jose. Pdf annotator for moodle quiz. 2023. URL https://github.com/Parvathy-S-Kumar/Moodle_Quiz_PDF_Annotator. Accessed 14-May-2024.

[3] Tausif Iqbal and Vishal Rao. Quiz annotator plugin that enables teacher to annotate essay type question in moodle. 2022. URL https://github.com/TausifIqbal/moodle_quiz_annotator. Accessed 14-May-2024.

[4] Moodle Developer Resources. Documentation. URL https://moodledev.io/docs. Accessed 14-May-2024.

[5] Moodle Developer Resources and Documentation. Plugin contribution checklist. URL https://moodledev.io/general/community/plugincontribution/checklist. Accessed 14-May-2024.

[6] Moodle Pty Ltd. Running acceptance tests. URL https://moodledev.io/general/development/tools/behat/running. Accessed 14-May-2024.

Thank you