Pacific Journal of Technology Enhanced Learning, 2024, 6(2).

The Design and Implementation of PAGE: Personalised Assessment Generative Engine

Johnny Chan
University of Auckland

Songyan Teng
University of Melbourne

The adoption of online learning has introduced a multitude of opportunities and complexities. This shift poses a challenge in preserving the integrity of digital assessments, which is further accentuated by the increasing accessibility of online resources. Assessment design has emerged as a promising solution to tackle this issue, with a specific focus on creating assessments that are resistant to cheating behaviour. Personalised assessments, in particular, have shown promise in reducing academic dishonesty. This work introduces the Personalised Assessment Generative Engine (PAGE); it is designed and implemented to simplify the process of creating and administering personalised digital assessments for the Canvas Learning Management System (LMS). Following the design science research methodology, PAGE has been developed to efficiently generate personalised assessments with variations in questions, answers, and additional materials. Deploying PAGE in a university course uncovered several benefits and considerations for educators with using such a tool in their classrooms. The strengths and weaknesses of PAGE are analysed, highlighting its application areas and potential avenues for future work.

Keywords: Personalised learning; Assessment creation; Assessment design; Educational technology

Introduction

In the ever-evolving education field, the adoption of online learning has presented both unprecedented opportunities and challenges. Leveraging technology to facilitate digital assessments significantly enhances their scalability and reach. However, a pressing challenge educators face is ensuring the integrity of digital assessments (Adedoyin & Soykan, 2020). Assessment design has been suggested as a promising method to address this challenge, where a focus is placed on designing assessments in ways that are less prone to cheating behaviour (Baird & Clare, 2017). In particular, assessment personalisation has been demonstrated as an effective strategy for reducing academic dishonesty (Sullivan, 2016). Utilising personalised assessments, each student or small group of students would receive a varied assessment version from others, and they cannot directly copy the answers or computation methods from their peers.

Although personalised digital assessments could enhance scalability and minimise cheating, implementing them in practice can be difficult when teachers also need to generate, deploy, and manage them efficiently, particularly for large student cohorts. We aim to address this problem by developing Personalised Assessment Generative Engine (PAGE), a tool designed and implemented to facilitate the creation and deployment of personalised assessments for the Canvas Learning Management System (LMS). In this paper we discuss the design, implementation, and evaluation of PAGE through a design science research lens, demonstrating how this tool could simplify and automate the creation and distribution of personalised digital assessments. By examining the technical detail of PAGE's integration within the Canvas LMS and assessing its real-world implementation, we provide insights into the practical solutions it offers to educators seeking to achieve both assessment personalisation and efficiency. We explore the challenges and considerations of deploying the tool, allowing educators to assess how it can be adapted for their purposes.

Literature Review

Assessment Scalability and Academic Dishonesty

Assessment scalability in education involves efficiently evaluating the learning outcomes of numerous students in an educational system. As education evolves globally, the significance of scalability becomes more apparent, emphasising the need for streamlined assessment processes to accommodate growing student numbers. Compared to conventional assessments, online assessments excel in scalability and accommodate diverse learning and teaching requirements (Veale & Craig, 2022). Additionally, they could provide feedback to many students via automated answer verification techniques (Naranjo et al., 2019).

However, online education presents several challenges, notably in upholding assessment integrity. There has been a surge in academic dishonesty amid the COVID-19 pandemic with the absence of on-campus invigilation (Turnbull et al., 2021). Online settings make collaborative cheating easier, allowing students to share answers discreetly. The nature of online assessments also makes monitoring closed-book online exams more difficult. Students can easily access unauthorised resources to trivialise the assessments (Newton & Essex, 2022). Consequently, many institutions have shifted to open-book formats, allowing resource access while designing assessments to discourage over-reliance. Yet, this shift may affect the core purpose of certain assessments, such as focusing quantitative evaluations on written explanations instead of technical rigour.

Mitigating Academic Dishonesty in Online Assessment

Traditional cheating deterrence involves in-person supervision during assessments. Although this is not possible with online assessment, various methodologies and tools have been implemented to imitate in-person supervision. Alessio et al. (2017) explored using proctoring software while students were taking their online tests, capturing students' desktops and surroundings. Students who sat proctored tests were found to use significantly less time but scored less compared to those who were un-proctored, suggesting that the proctored environment may have reduced the presence of cheating. However, these video recordings must be manually reviewed, which hampers scalability. AI solutions have been introduced for automatically classifying suspicious behaviour in online assessments and notifying the proctor (Saba et al., 2021). Although these solutions reduce manual checking, the level of monitoring may negatively impact the wellbeing of students (Barrett, 2021).

Mitigating cheating via resource restriction is a common strategy, including locking browsers to specific pages to limit communication with other students and resource access (Fluck, 2019). However, ensuring that students do not use other undetected devices to cheat remains difficult. Specific question types for online assessments have been proposed to deter plagiarism. One strategy is to create assessments centred on higher-order thinking, such as assessing critical analysis or summarisation skills, which is considered to be difficult for cheating (Sotiriadou et al., 2019). Assessments requiring creating new information, including suggestions for specific scenarios, or making predictions, allow for evaluating individual thought. Although these assessment types may help to reduce cheating by providing a large answer space, the variation in student answers means that they need to be manually verified. Requiring to manually check through complex answers could pose difficulties for scaling assessments. The design of assessment delivery has also been used to discourage plagiarism. A common approach is through non-uniform assessments (Alin et al., 2022), which may include selecting questions from a large question bank or introducing variations within each question. Randomising question order for each student generates assessment variation, making it more difficult for direct copying between students (Chirumamilla et al., 2020). However, this level of variation is minor as students still receive the same questions as their peers, and they can simply copy the answers of others in a different order.

Personalised Assessment

Personalised assessments have been widely used to target assessment design for cheating mitigation, and is amongst the best assessment types for doing so (Bretag et al., 2018). Valizadeh (2022) suggested creating unique assessments for every student, such that every student would receive a different assessment variation. Individualised Excel-based exams have been proposed as a method to prevent cheating in online exams (Suryani, 2020). By providing each student with a unique exam, it becomes more challenging for students to collaborate or copy answers from one another. Manoharan (2019) applied personalisation to multiple-choice assessments, using an HTML format with macros to structure and create randomised questions. Student feedback showed they were in favour of personalised tests and generally believed that personalisation reduced the level of cheating. When evaluated with a group of staff members, they found that it was not possible to cheat in these personalised tests unless they were allowed to have discussions with others. In a similar study, where a programmable framework to create personalised assessments was deployed and evaluated in a large course, more than 70% of the students agreed that personalised assignments motivated them to engage in independent work and reduced plagiarism (Manoharan, 2017). Changing certain parameters to create personalised questions has also been explored, in which the wording of questions and variable values can be altered to create different question versions (Boland & Jacobs, 2014). Using this personalised assessment approach within an accounting course, students perceived that the occurrence of cheating on assessments decreased as the course progressed over the semester. Generating unique datasets for a data analytics test in a chemistry course has also been trialled, suggesting that students would be less likely to cheat as a result of this personalisation approach due to them needing to do the work again with a different dataset if they intend to cheat with their peers (Grove, 2022).

Although personalised assessments can greatly improve assessment uniqueness across students and reduce cheating, manually creating these variations is laborious and difficult for large course sizes. There is also a lack of tools that can effectively integrate personalised assessments and supplementary material, such as datasets, within existing learning platforms. Thus, a tool is necessary for facilitating assessment creation and distribution in a scalable environment. To address this, we design and develop PAGE which efficiently generates personalised assessments. We focus on enhancing assessment scalability while aiming to mitigate academic dishonesty. We detail our engine's design, implementation, and application in the subsequent sections.

Research Methodology

Online assessments are scalable for different requirements (Veale & Craig, 2022). With selected learning platforms and LMS offering automated scoring algorithms, they also enable automation in grading and providing feedback. Immediate feedback helps students to identify areas for improvement and adjust their learning strategies (Hooda et al., 2022). However, online assessments are also prone to cheating. As mentioned before, several methods have been developed to tackle academic dishonesty in online assessment but there are only a few explorations into how we can mitigate the issue through assessment design. To address this gap, we followed the design science research methodology (Peffers et al., 2007) to develop a tool facilitating the creation and deployment of personalised digital assessments.

We initially encountered the need for scalability in several undergraduate business analytics courses taught at our university. From our ideation process, we explored delivering online assessments to accommodate our growing course sizes. We noted the challenges of requiring technical infrastructure that could support a large cohort of students and methods for mitigating cheating through assessment design. To address them, we developed a prototype aimed at making assessment creation more streamlined. We designed our assessments to be personalised, utilising our engine to create variation in our assessment questions while maintaining a consistent level of difficulty between assessment variations. To evaluate our engine's practicality, we applied it to the assessments in one of our business analytics courses over multiple semesters. Finally, the teaching team evaluated the engine by comparing the efficiency and cost of assessment creation across each semester.

The Design, Implementation and Evaluation of PAGE

We applied the design science research methodology (DSRM) process model by Peffers et al. (2007) to develop PAGE, as indicated in Figure 1.

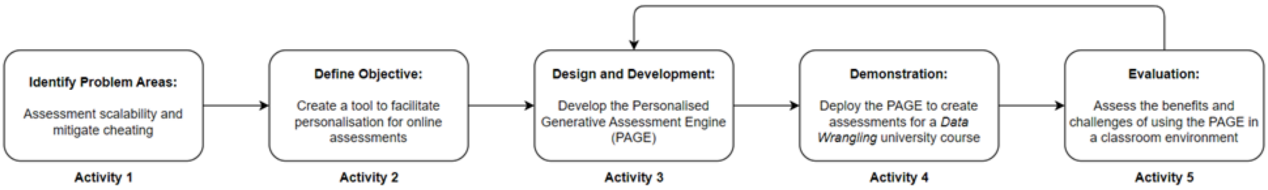

Figure 1: The Research Framework for PAGE

In Activity 1, we identified the need for scalability in business analytics courses, prompted by increasing course sizes. We identified online assessment as a means for scalability, and we sought to develop a strategy for designing online assessments that could better address cheating.

In Activity 2, we specified what our solution aims to achieve. We employed creating assessment versions as a form of personalisation to mitigate cheating. For each assessment, multiple versions were created with variations in their questions and answers. Assessments requiring additional material such as datasets can also randomise these materials for each version, further enhancing randomness. However, variations in the questions should be minor, such as modifying variable values, to ensure that difficulty remains consistent across each version. By implementing assessment variation such that every student receives a different but comparable version, we aimed to reduce the possibility of direct method or solution-copying. To achieve this, we required a tool to facilitate online assessment creation. Designing multiple assessment versions and manually creating them online for distribution to students is time-consuming and may not scale well with larger courses. Therefore, we needed an engine to streamline this process and to allow for batch-creating assessments.

In Activity 3, we discussed how we should design and develop PAGE. We first decided to integrate with the Canvas LMS used in our university. We investigated working with the Canvas API. We used a Node.js environment with the Express framework to build our engine with efficient access to web APIs. The assessment generation process is divided into three stages: 1) dataset generation, 2) question-and-answer generation, and 3) quiz creation. We outline each stage below, as shown in Figure 2.

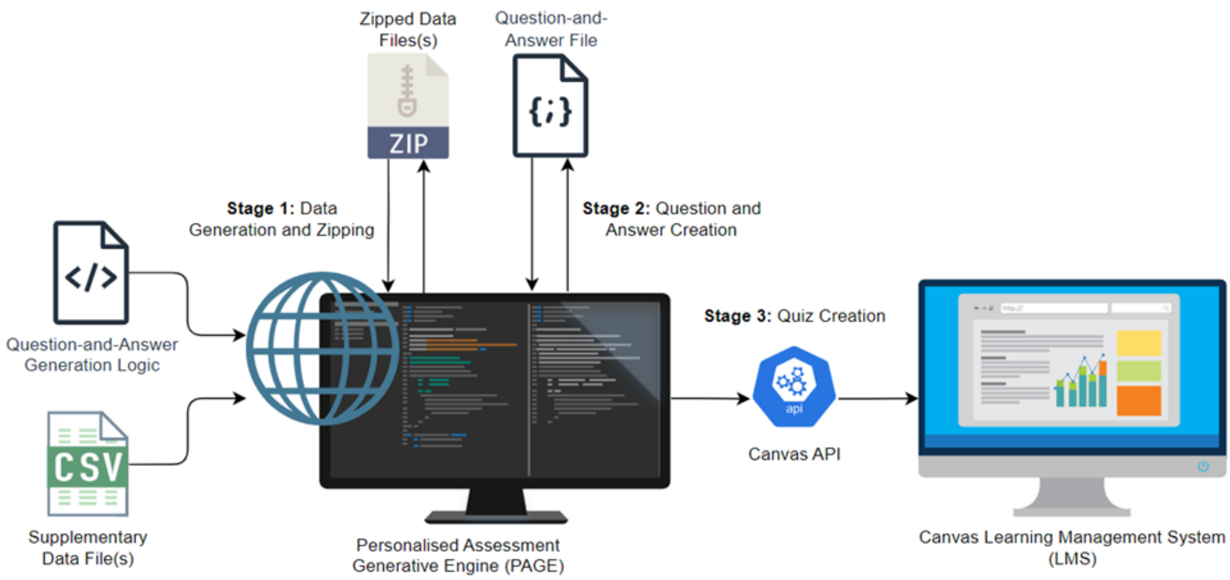

Figure 2: The System Architecture of PAGE

Stage 1: Dataset Generation - The dataset generation stage introduces the first variation step for assessments requiring supplementary resources such as datasets. In our case, this stage takes a master dataset as input and produces subsets of data as output. Instructors can specify how large they require each data subset to be, and the engine selects a random subset from the master dataset for use in each assessment version. For example, we could use an Excel file with 10,000 entries as the master dataset to create ten unique data subsets with approximately 9,000 randomly selected entries each. Creating a unique dataset for each assessment version means that even the same questions will have different answers between the assessment versions. This aims to reduce plagiarism and discourage direct copying of answers between peers, as well as provide an additional avenue of personalisation beyond assessment questions. Once the data subsets have been generated, the engine compresses (i.e., zips) and uploads them to a specified cloud storage location (e.g., Canvas LMS). These files are visible to the instructor, as shown in Figure 3.

Figure 3: Zipped Data Folders Available for Student Downloading in Canvas LMS

Stage 2: Question-and-Answer Generation - The question-and-answer generation stage creates questions and uses the generated data subsets to compute corresponding answers. This stage implements the second variation step, where each question may receive modifications across different assessment versions. In addition to generating multiple data subsets, creating variation in the questions further aids in personalisation as each question variation requires different approaches. Instructors can use Node.js functionalities relevant to their assessment to create scripts that compute an answer for each question. Doing so commonly includes utilising Node.js packages designed to work with specific technologies (e.g. the XPath package for JavaScript for working with XPath; the MongoDB Node Driver for working with MongoDB). Because instructors need to provide methods for solving each question via scripts, it is expected that users of the engine are comfortable with programming.

Figure 4 shows an example of a script for creating a question based on a video games dataset in CSV format, asking how many video games contain a particular phrase in their name. The script calculates this answer and stores this in the answer variable. The phrase is randomly selected out of four options: Pokémon, Wii, Super Mario, and Grand Theft Auto, creating four variations of one question. Instructors can decide how many versions they require for each question and make adjustments accordingly. When combined with multiple other questions, each having its own variations in a complete assessment, it is highly unlikely that any two assessment versions will share identical questions.

Functionality is also provided to create questions and answers for assessments without supplementary materials, such as for some mathematics assessments. Figure 5 presents a script for creating a mathematics question, asking for the value of 8x+y, where x is a random integer from 1 to 20, and y is a random integer from 1 to 5. By varying questions using this method, instructors can easily create 100 different versions of a simple question that can be randomly selected with minimal effort. After creating the questions and answers for each version, they are combined and stored in a JSON file for publication to Canvas, which is shown in Figure 6.

```
/**
 * @param {number} hash
 * @param {array} csv
 * @param {function} questionDone
 */
function two(hash, csv, questionDone) {
  const qNum = 2;

  const names = ["Pokemon", "Wii", "Super Mario", "Grand Theft Auto"];

  const randNum = Math.floor(Math.random() * names.length);

  const name = names[randNum];

  let numValues = 0;

  for (let i = 1; i < csv.length; i++) {
    let row = csv[i];
    if (row[util.header.Name].includes(name)) {
      numValues++;
    }
  }

  studentQA[hash][
    `q${qNum}q`
  ] = `How many video games contain "${name}" in their name?`;
  const answer = numValues;
  studentQA[hash][`q${qNum}a`] = `${answer}`;

  questionDone(null);
}
```

Figure 4: Sample Question Script Using a CSV File

```javascript
/**
 * @param {number} hash
 * @param {array} csv
 * @param {function} questionDone
 */
function four(hash, csv, questionDone) {
  const qNum = 4;

  const randX = Math.ceil(Math.random() * 20);
  const randY = Math.ceil(Math.random() * 5);

  const solution = 8 * randX + randY;

  studentQA[hash][
    `q${qNum}q`
  ] = `What is 8x + y, where x = ${randX} and y = ${randY}?`;
  const answer = solution;
  studentQA[hash][`q${qNum}a`] = `${answer}`;

  questionDone(null);
}
```

Figure 5: Sample Question Script Using No Dataset

```json
{
  "2wKGZ$FWV": {
    "q1q": "For video games released in the year \"1989\", how many more sales units were made in North America compared to Japan?",
    "q1a": "26.79",
    "q2q": "How many video games contain \"Wii\" in their name?",
    "q2a": "52",
    "q3q": "What are the total global sales for games that are <strong>not</strong> published by Nintendo?",
    "q3a": "7080.72",
    "q4q": "What is 8x + y, where x = 18 and y = 5?",
    "q4a": "149"
  },
  "T61RxQfJvw": {
    "q1q": "For video games released in the year \"1997\", how many more sales units were made in North America compared to Japan?",
    "q1a": "45.72",
    "q2q": "How many video games contain \"Pokemon\" in their name?",
    "q2a": "35",
    "q3q": "What are the total global sales for games that are <strong>not</strong> published by Nintendo?",
    "q3a": "7113.35",
    "q4q": "What is 8x + y, where x = 9 and y = 4?",
    "q4a": "76"
  },
  "8TTgAQv0xO": {
    "q1q": "For video games released in the year \"2001\", how many more sales units were made in North America compared to Japan?",
    "q1a": "133.44",
    "q2q": "How many video games contain \"Wii\" in their name?",
    "q2a": "52",
    "q3q": "What are the total global sales for games that are published by Nintendo?",
    "q3a": "1781.53",
    "q4q": "What is 8x + y, where x = 11 and y = 4?",
    "q4a": "92"
  }
}
```

Figure 6: JSON File for Storing Questions and Answers for each Assessment Version

Stage 3: Quiz Creation - The quiz creation stage populates quizzes in Canvas using the generated questions and answers and randomly allocates these quizzes to students. Assessment parameters such as start and due dates, number of attempts, and the number of available marks are set at this stage, removing the need to apply these settings via the Canvas user interface. The engine processes these settings and utilises the Canvas API to create unique versioned assessments and make them available for each student. Figure 7 shows how the generated assessments are displayed in the Canvas assignment view, and Figure 8 contains the details of one assessment version, demonstrating how the questions provided in the JSON file are presented on Canvas.

Figure 7: Generated Assessment Versions in Canvas LMS

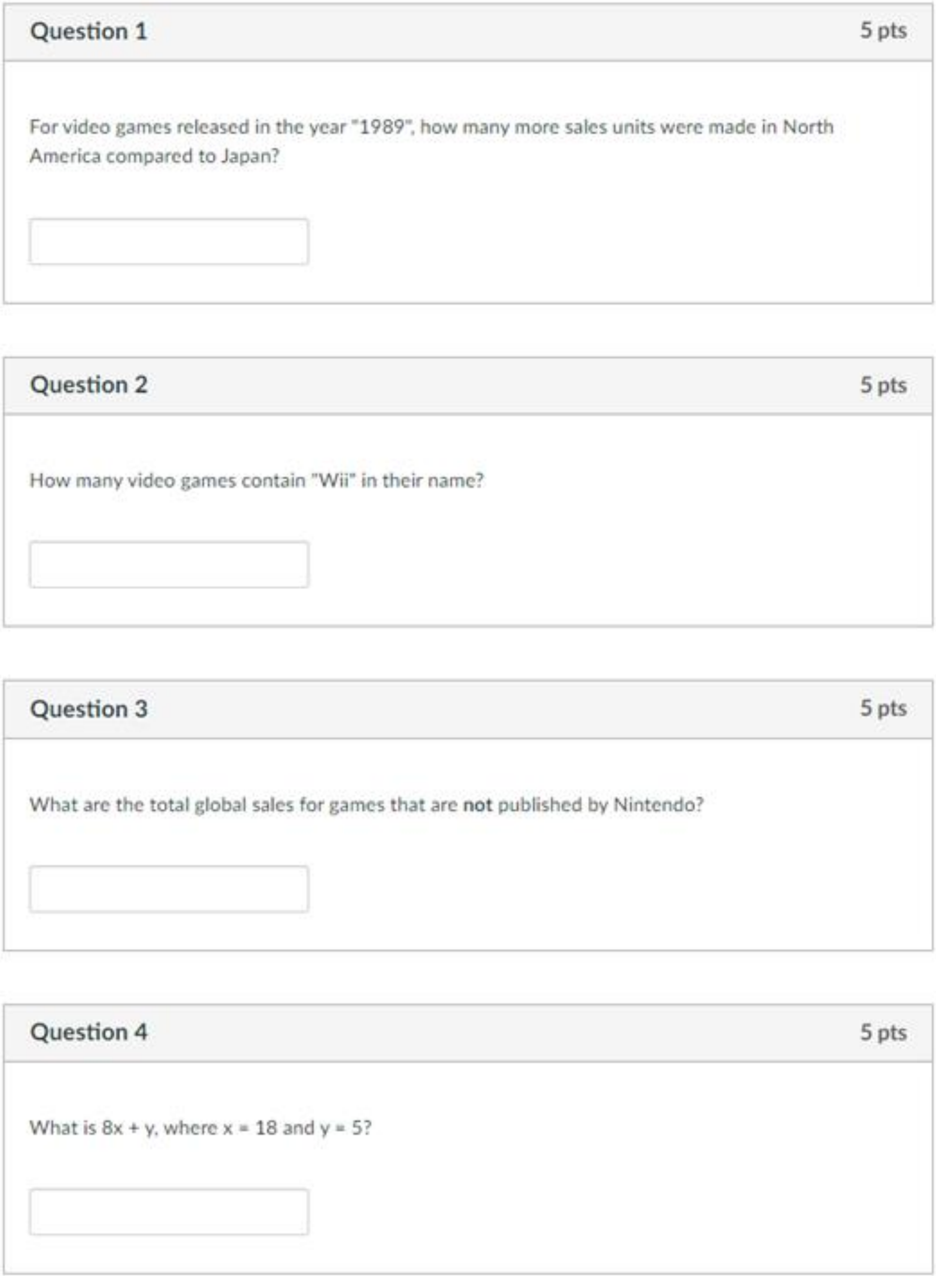

Figure 8: Assessment Questions for one Assessment Version in Canvas LMS

In Activity 4, we used PAGE for assessment creation in our Data Wrangling course. Students are exposed to different formats of big data and methods for working with them, such as Microsoft Excel for CSV files, XPath for XML files, and MongoDB for NoSQL databases. It aims to enhance students' technical skills while promoting rigour and self-reflection in their work. Assessments in the course involve calculation-based tasks where students manipulate provided data to derive a single answer. Students are allocated multiple attempts for each assessment, with each attempt providing immediate feedback on the correctness of each question. We designed the questions to remain the same across each attempt, allowing students to learn from errors and approach the questions differently in subsequent attempts. Using PAGE, we design questions and develop code for computing their corresponding answers. We also specify assessment settings, including due dates, number of attempts, and the total grade, which is passed to the engine for populating them to Canvas. Each unique assessment is then assigned to the corresponding student.

In Activity 5, we evaluated the engine. We assessed the feasibility of hosting versioned assessments on the Canvas LMS. Facilitating traditional online assessments, where each student completes the same assessment questions, requires only one quiz that all students access and complete. However, versioned assessments require a unique quiz to be created for each student so they can be granted the correct access permissions, such that only they can view and attempt the quiz. In the case of a 100-student course, there is a need to generate 100 unique quizzes for each assessment, as opposed to just one in traditional assessments. These implications affect Canvas's loading times and storage capacity. Extended load times and increased storage can hinder instructor productivity. We tested the load time for displaying all assessments in our course of 100 students. We first used the traditional assessment approach, where assessments are not versioned and therefore require only one quiz distributed across the course. Testing with the seven assessments in our course, Canvas loaded and displayed all assessments within approximately one second. We then compared this to versioned assessments, where each assessment is generated uniquely for each student, resulting in a total of 700 quizzes. Canvas took approximately three minutes to load and display all assessments, which is over 100 times longer than that of the non-versioned assessments. This indicates that the growth in loading time may exceed the increase in course size, possibly due to added overhead in fetching large data volumes. Because teachers frequently access assessments for viewing and editing, this excess in time can be costly. Additionally, assessments involving supplementary materials, like datasets, require generating a unique set of materials for each student. Designing such assessments for large student cohorts can lead to substantial storage costs. These factors suggest that achieving full personalisation in assessments might not be as scalable for larger courses in terms of infrastructure and costs.

Based on our evaluation, we iterated back to Activity 3 to improve the flexibility of our engine. Although creating a unique assessment for every student achieves the highest level of personalisation, it is worth considering the option of assigning multiple students to the same assessment version without compromising our cheating mitigation goal. For example, generating 50 assessment versions and distributing them evenly across 100 students results in every version being assigned to two students. This approach reduces the required storage by half and cuts loading times by more than half. We argue that distributing each assessment version among a small group of students has a minimal impact on the level of cheating mitigation. Therefore, we built additional functionality allowing instructors to specify how many assessment versions to generate, which are then distributed evenly and randomly to students. In doing so, we provide instructors with greater control over assessment design, allowing them to create more assessment versions for a greater level of personalisation at the expense of time and storage, or create fewer versions to reduce waiting time and storage size.
Upon completing this redesign, we again applied PAGE in a classroom setting in Activity 4. This was done in the semester following our first deployment, again with the same course. We followed the same methodology as our first trial, though we created only 10 versions of each assessment for a course of 100 students, rather than a unique version for each student.

We then evaluated our revised approach in Activity 5. Testing again with the seven assessments in our course, the LMS loaded and displayed all assessments in approximately 10 seconds, which is 18 times less than the time required for loading fully unique assessments for a course of the same size. Similarly, hosting our datasets required approximately 10 times less storage. While the level of assessment personalisation may be somewhat reduced, this approach strikes a balance between scalability and maintaining the crucial elements of personalisation needed to deter cheating.

Discussion

The implementation of PAGE in a tertiary education setting allowed us to understand the considerations, challenges, and benefits of deploying such tool. We discuss our findings in the following sections.

Assessment Type

Assessments in our Data Wrangling course are primarily calculation-based, where questions can be built from variables and altered at large. These may include asking for various orderings (e.g., find the second highest vs. find the fifth highest), initialising different variables (e.g., suppose X = 1000 vs. suppose X = 5000), or changing conditionals (e.g., how many weights are below 20 grams vs. how many weights are below 50 grams). Since these questions maintain a consistent structure and are altered only in their sub-components, it becomes possible to generate numerous versioned assessments and calculate their corresponding answers using the engine. This can be achieved by adding automation to altering these variables, such as incorporating randomness using JavaScript's built-in random number generator. For calculation-based questions requiring only a single answer, grading can be completed automatically within Canvas. This means that manual grading is unnecessary, regardless of the number of assessment versions deployed.

However, this method is less applicable to qualitative assessments such as essays, where questions vary greatly in content and format. On top of that, grading essays involves assessing not just the correctness of answers but also the quality of argumentation, depth of analysis, and clarity of expression (Suhartono et al., 2020). These nuanced aspects require human expertise and judgement, making it challenging to fully automate the grading process (Shermis et al., 2010). Grading multiple essay versions also presents several challenges, including the need for grading consistency across variations, varied expertise across different essay types, and allocating additional time and resources to provide a fair assessment and meaningful feedback (Roscoe & McNamara, 2013). These considerations make delivering and grading qualitative assessments more difficult to scale.

Despite these challenges, machine learning techniques can be applied to support the scalability of qualitative assessments (Chang et al., 2021). Natural language processing (NLP) algorithms can assist educators in tasks such as automated scoring, content analysis, and summarisation (Hussein et al., 2019). For example, automated scoring systems can provide preliminary assessments. They can save time for graders by first evaluating written work and providing an estimated score, then highlighting sections requiring human evaluation. They can also be used to analyse the structure of essays, helping teachers gain insights into common strengths and weaknesses across their student cohort.

Infrastructure

One of the constraints we encountered was the inherent structure of Canvas. While it is a robust and widely used platform for course management and content hosting, its assessment delivery capabilities are targeted towards conventional assessments. These assessments typically have a fixed format with a small number of quizzes distributed to all students and therefore lack the flexibility required to support versioned assessments. Because there are no automated methods for altering quiz settings to allow for creating a large number of assessment versions, each version requires a unique quiz to be created and distributed through PAGE. The large number of quizzes required can greatly increase their loading time, making it more difficult for instructors to view and edit assessments.

Storing assessment supplementary files (e.g., datasets) can also be challenging for versioned assessments. Many institutions impose restrictions on storage for cloud-based storage providers such as Google Drive, Microsoft OneDrive, and Dropbox. In our case, we stored all files within Canvas and linked these files to their associated quizzes. The files are only accessible via their URL, such that they are not directly visible to other students in the course (Instructure, 2023). However, although each uploaded file on Canvas has a unique ID that can be accessed within its URL, files that are simultaneously uploaded will have consecutive IDs. Therefore, students can potentially guess and alter the ID within the URL to access other students' files, as there are no existing options in Canvas to alter the visibility of files individually for each student. This may lead to unfair assessment as students can access a greater pool of data to test with than what they're permitted. Similar to other cloud storage providers, Canvas provides limited file storage capacity (Instructure, 2022). Hence, the number of versions to be created for each assessment may be constrained by the size of its supplementary files.

The limitations we encountered within Canvas highlighted the need for a bespoke platform tailored to support versioned assessments. The platform should address optimisation challenges, such as removing the need to create entirely unique quizzes for each assessment version. Instead, it could allow instructors to customise and randomise specific elements within each question of a quiz while preserving its core structure. Although Canvas facilitates using question banks and random question ordering to create variation in assessments, there are no efficient ways to do this on a large scale and with the level of variation generated by PAGE. Additionally, this platform should offer enhanced data management capabilities, providing educators with an efficient means of storing multiple large files associated with different assessment versions. It should also implement robust access controls to restrict unauthorised student access, ensuring that materials are visible only to the corresponding students.

Ease of Use

Creating assessments using PAGE requires instructors to have programming proficiency, particularly in JavaScript. In addition, instructors must also be able to translate their domain expertise practically in a code format utilising relevant packages. This involves crafting tailored logic for each question that guides how the engine computes answers for different question variations. Although doing so may be straightforward for programming-based assessments that already use a code environment, creating assessments that are not programming-based may be inherently more difficult. As there is currently no available graphical user interface for our engine, instructors must provide the logic for these computations using raw Node.js functionality, such as using regular expressions in JavaScript for data matching. Translating problem-solving methods across platforms may be challenging, requiring careful planning and feasibility assessment. Although PAGE greatly reduces the time required for generating and uploading versioned assessments compared to manual methods, we acknowledge that the process of creating questions and formulating the logic to compute answers can still be time-intensive.

The need for specific packages and programming techniques for different assessment topics means that the code developed for one assessment is less reusable across various topics. Each new assessment often necessitates unique approaches and coding solutions, as requirements can vary significantly from one subject area to another. While this tailored approach enhances the tool's adaptability to diverse assessment domains, it also contributes to the time investment required in designing and coding assessments. For example, creating a versioned statistics assessment involving calculations of simple statistical measures may require less code logic, as calculations can be directly translated into code format and executed using JavaScript functions. However, another assessment for the same statistics course may require using the R programming language, which would necessitate a different implementation with PAGE due to the assessment itself needing distinct tools and computation methods.

This challenge potentially could be addressed by developing a repository of reusable code snippets and templates that encapsulate common assessment logic. Instructors could customise examples from this repository to streamline the creation of assessments for similar topics or to expedite the process for common assessment types.

Generalisation to Education

One aspect of PAGE's generalisation potential is its applicability in Massive Open Online Courses (MOOCs). MOOCs have gained large popularity as a means of online education due to their convenience, and they often require scalable assessment solutions. By integrating PAGE into MOOC platforms, educators can leverage its ability to generate personalised assessments for a massive and diverse online audience. The tool's capacity to generate question variations and personalised supplementary files is well-suited to the dynamic and expansive nature of MOOCs. Similarly, PAGE could be beneficial in enhancing professional certification assessments. Many certification programs involve rigorous assessments and personalised versions of these assessments can add an extra layer of security to the certification process.

In addition to large-scale assessments, PAGE can also be used for personal learning. Individuals seeking self-assessment can harness the tool to repeatedly create new assessment versions for use as revision or mock exams. Instructors can also utilise PAGE to craft customised practice exams that mimic the conditions and complexity of upcoming assessments, which can be distributed to students. The tool's ability to generate personalised questions ensures that each practice test is unique, promoting a thorough understanding of the topic. This feature is particularly beneficial when studying for standardised tests, certification exams, or any assessment that requires preparation across a range of scenarios.

Each application area, whether in a university course, a MOOC, or individual self-assessments, requires some calibration to align with specific objectives and standards. Nevertheless, PAGE's versatility offers a flexible framework adaptable to a wide array of assessment types, serving as a valuable tool for creating engaging and effective learning experiences.

Future Work

The effectiveness of assessments created using PAGE should be further explored through more student and instructor evaluations from various domains. This includes collecting feedback to understand the perception of its impact on cheating mitigation, assessment fairness, and the overall learning experience.
Currently, utilising PAGE requires instructors to have programming knowledge. Therefore, developing an intuitive user interface to facilitate question-and-answer creation could help to streamline this process further. This can help to reduce the complexity of using our engine, making creating personalised assessments accessible to a broader spectrum of educators.

The rapid advancement of generative AI (GenAI) has been adopted in education to enhance personalised learning and create tailored learning experiences for students (Hashim et al., 2022). Combining PAGE with GenAI has the potential to augment personalised assessments by automating the generation of tailored assessment questions. GenAI algorithms, driven by extensive datasets, can create questions that adapt to individual student learning preferences and proficiency levels (Das et al., 2023). Additionally, this integration can enable PAGE to provide personalised feedback and learning resources in real time, offering a dynamic and adaptive learning experience (Su & Yang, 2023). Furthermore, by using historical student data, GenAI can facilitate the continuous improvement of personalised assessments, refining personalisation strategies based on past student performance (Iffort, 2023).

Conclusion

In this paper, we demonstrate the development and implementation of PAGE as a practical solution to address the challenges of online assessments in the context of scalability and academic integrity. By applying the design science research methodology, we created a tool that streamlines the process of generating and deploying personalised digital assessments. Through our iterative design and development process, we show how PAGE can efficiently create assessments with variations in questions and answers, as well as additional assessment material such as datasets. To assess PAGE in a classroom environment, we deployed the tool to generate assessments in one of our university's courses. We identify several benefits and considerations associated with our tool, and we provide suggestions for addressing current challenges. We believe the development of PAGE presents an important step towards achieving more scalable and personalised assessments, ultimately enhancing the quality of learning experiences for both students and educators.

References

Adedoyin, O. B., & Soykan, E. (2020). Covid-19 pandemic and online learning: The challenges and opportunities. Interactive Learning Environments, 31(2), 863–875. https://doi.org/10.1080/10494820.2020.1813180

Alessio, H. M., Malay, N. J., Maurer, K., Bailer, A. J., & Rubin, B. (2017). Examining the Effect of Proctoring on Online Test Scores. Online Learning, 21(1). https://doi.org/10.24059/olj.v21i1.885

Alin, P., Arendt, A., & Gurell, S. (2022). Addressing cheating in virtual proctored examinations: Toward a framework of relevant mitigation strategies. Assessment & Evaluation in Higher Education, 48(3), 262–275. https://doi.org/10.1080/02602938.2022.2075317

Baird, M., & Clare, J. (2017). Removing the opportunity for contract cheating in business capstones: A crime prevention case study. International Journal for Educational Integrity, 13(1). https://doi.org/10.1007/s40979-017-0018-1

Barrett, L. (2021). Rejecting Test Surveillance in Higher Education. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3871423

Boland, C., & Jacobs, F. (2014). How Personalized Assignments Can Increase Learning and Reduce Cheating in a Large-Lecture Introductory Accounting Class. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2413195

Bretag, T., Harper, R., Burton, M., Ellis, C., Newton, P., Haeringen, K. van, Saddiqui, S., & Rozenberg, P. (2018). Contract cheating and assessment design: Exploring the relationship. Assessment & Evaluation in Higher Education, 44(5), 676–691. https://doi.org/10.1080/02602938.2018.1527892

Chang, L.H., Rastas, I., Pyysalo, S., & Ginter, F. (2021). Deep learning for sentence clustering in essay grading support. arXiv. https://doi.org/10.48550/ARXIV.2104.11556

Chirumamilla, A., Sindre, G., & Nguyen-Duc, A. (2020). Cheating in e-exams and paper exams: The perceptions of engineering students and teachers in Norway. Assessment & Evaluation in Higher Education, 45(7), 940–957. https://doi.org/10.1080/02602938.2020.1719975

Das, A., Malaviya, S., & Singh, M. (2023). The Impact of AI-Driven Personalization on Learners' Performance. International Journal of Computer Sciences and Engineering, 11, 15–22. https://doi.org/10.26438/ijcse/v11i8.1522

Fluck, A. E. (2019). An international review of eExam technologies and impact. Computers & Education, 132, 1–15. https://doi.org/10.1016/j.compedu.2018.12.008

Grove, J. (2022). Could Custom Exams Prevent Cheating? https://www.insidehighered.com/news/2022/01/06/british-university-tries-custom-exam-reduce-cheating

Hashim, S., Omar, M. K., Jalil, H. A., & Sharef, N. M. (2022). Trends on Technologies and Artificial Intelligence in Education for Personalized Learning: Systematic Literature Review. International Journal of Academic Research in Progressive Education and Development, 11(1). https://doi.org/10.6007/ijarped/v11-i1/12230

Hooda, M., Rana, C., Dahiya, O., Rizwan, A., & Hossain, M. S. (2022). Artificial Intelligence for Assessment and Feedback to Enhance Student Success in Higher Education. Mathematical Problems in Engineering, 2022, 1–19. https://doi.org/10.1155/2022/5215722

Hussein, M. A., Hassan, H., & Nassef, M. (2019). Automated language essay scoring systems: A literature review. PeerJ Computer Science, 5, e208. https://doi.org/10.7717/peerj-cs.208

Iffort. (2023). Exploring the Role of Generative AI in Personalised Learning and Curriculum Development. https://www.iffort.com/blog/2023/04/13/generative-ai-role-in-personalised-learning

Instructure. (2022). Canvas File Quotas. https://community.canvaslms.com/t5/Canvas-Resource-Documents/Canvas-File-Quotas/ta-p/387061

Instructure. (2023). Uploading Files. https://canvas.instructure.com/doc/api/file.file_uploads.html

Manoharan, S. (2017). Personalized Assessment as a Means to Mitigate Plagiarism. IEEE Transactions on Education, 60(2), 112–119. https://doi.org/10.1109/te.2016.2604210

Manoharan, S. (2019). Cheat-resistant multiple-choice examinations using personalization. Computers & Education, 130, 139–151. https://doi.org/10.1016/j.compedu.2018.11.007

Naranjo, D. M., Prieto, J. R., Moltó, G., & Calatrava, A. (2019). A Visual Dashboard to Track Learning Analytics for Educational Cloud Computing. Sensors, 19(13), 2952. https://doi.org/10.3390/s19132952

Newton, P., & Essex, K. (2022). How common is cheating in online exams and did it increase during the COVID-19 pandemic? A Systematic Review. https://doi.org/10.21203/rs.3.rs-2187710/v1

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/mis0742-1222240302

Roscoe, R. D., & McNamara, D. S. (2013). Writing pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. Journal of Educational Psychology, 105(4), 1010–1025. https://doi.org/10.1037/a0032340

Saba, T., Rehman, A., Jamail, N. S. M., Marie-Sainte, S. L., Raza, M., & Sharif, M. (2021). Categorizing the Students' Activities for Automated Exam Proctoring Using Proposed Deep L2-GraftNet CNN Network and ASO Based Feature Selection Approach. IEEE Access, 9, 47639–47656. https://doi.org/10.1109/access.2021.3068223

Shermis, M. D., Burstein, J., Higgins, D., & Zechner, K. (2010). Automated Essay Scoring: Writing Assessment and Instruction. In International Encyclopedia of Education (pp. 20–26). Elsevier. https://doi.org/10.1016/b978-0-08-044894-7.00233-5

Sotiriadou, P., Logan, D., Daly, A., & Guest, R. (2019). The role of authentic assessment to preserve academic integrity and promote skill development and employability. Studies in Higher Education, 45(11), 2132–2148. https://doi.org/10.1080/03075079.2019.1582015

Su, J., & Yang, W. (2023). Unlocking the Power of ChatGPT: A Framework for Applying Generative AI in Education. ECNU Review of Education, 6(3), 355–366. https://doi.org/10.1177/20965311231168423

Suhartono, D., Gema, A. P., Winton, S., David, T., Fanany, M. I., & Arymurthy, A. M. (2020). Argument annotation and analysis using deep learning with attention mechanism in Bahasa Indonesia. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00364-z

Sullivan, D. P. (2016). An Integrated Approach to Preempt Cheating on Asynchronous, Objective, Online Assessments in Graduate Business Classes. Online Learning, 20(3). https://doi.org/10.24059/olj.v20i3.650

Suryani, A. W. (2020). Individualized Excel-Based Exams to Prevent Students from Cheating. JABE (JOURNAL OF ACCOUNTING AND BUSINESS EDUCATION), 5(1), 14. https://doi.org/10.26675/jabe.v5i1.14367

Turnbull, D., Chugh, R., & Luck, J. (2021). Transitioning to E-Learning during the COVID-19 pandemic: How have Higher Education Institutions responded to the challenge? Education and Information Technologies, 26(5), 6401–6419. https://doi.org/10.1007/s10639-021-10633-w

Valizadeh, M. (2022). Cheating in Online Learning Programs: Learners' Perceptions and Solutions. Turkish Online Journal of Distance Education, 23(1), 195–209. https://doi.org/10.17718/tojde.1050394

Veale, A. J., & Craig, T. S. (2022). Design principles for final answer assessment in linear algebra: Implications for digital testing. Teaching Mathematics and Its Applications: An International Journal of the IMA, 41(4), 280–291. https://doi.org/10.1093/teamat/hrac002