Enabling Open Educational Resource Adoption through Integrated Sharing in PrairieLearn

Seth Poulsen
seth.poulsen@usu.edu
Utah State University
Logan, UT, USA

Geoffrey L. Herman
glherman@illinois.edu
University of Illinois
Urbana, IL, USA

Mariana Silva
mfsilva@illinois.edu
University of Illinois
Urbana, IL, USA

Max Fowler
mfowler5@illinois.edu
University of Illinois
Urbana, IL, USA

David H. Smith IV
dhsmith4@vt.edu
Virginia Tech
Blacksburg, VA, USA

Leo Porter
leporter@eng.ucsd.edu
University of California San Diego
La Jolla, CA, USA

Nico Ritschel
ritschel@illinois.edu
University of Illinois
Urbana, IL, USA

Craig Zilles
zilles@illinois.edu
University of Illinois
Urbana, IL, USA

Matthew West
mwest@illinois.edu
University of Illinois
Urbana, IL, USA

Abstract

This paper introduces the PrairieLearn Question Sharing System (PQSS), which enables instructors to share question generators with other instructors, either as open educational resources or privately. PQSS is integrated into PrairieLearn, an open-source, problem-driven online learning platform. PQSS addresses a critical need for more open-source assessments by making it easier for instructors to share assessments and for instructors to use those assessments. Instructors often do not share questions due to the time it takes to publish them and the lack of recognition for their work. Because it is directly integrated into PrairieLearn, PQSS reduces the aforementioned friction of sharing and using shared questions, and we can report usage statistics to help question authors receive recognition for their work. In this paper, we share design and implementation details of the system, as well as experiences using it to share course content across courses and between universities.

CCS Concepts

• **Applied computing** → **Learning management systems**.

Keywords

open educational resources, question sharing, model curricula

ACM Reference Format:
Seth Poulsen, Geoffrey L. Herman, Mariana Silva, Max Fowler, David H. Smith IV, Leo Porter, Nico Ritschel, Craig Zilles, and Matthew West. 2026. Enabling Open Educational Resource Adoption through Integrated Sharing in PrairieLearn. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2026), February 18–21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3770762.3772503

SIGCSE TS 2026, February 18–21, 2026, St. Louis, MO, USA

ACM ISBN 979-8-4007-2256-1/2026/02
https://doi.org/10.1145/3770762.3772503

1 Introduction

Learning, especially of procedural knowledge, requires students to perform tasks repeatedly. An abundance of research has shown that retrieval practice (when students articulate or use knowledge from memory) gives stronger learning gains than rehearsal practice (when students re-read or re-watch sources of information) [37, 41, 48–51]. Instructors can take advantage of the testing effect by using low-stakes testing in their pedagogies and their assessment strategies [25]. Active learning techniques (e.g., peer instruction [45], think-pair-share [55]) are forms of low-stakes testing. Similarly, assessment strategies such as low-stakes, frequent exams [19, 26, 31, 44], second-chance exams [17, 28, 33, 35], and mastery learning [22, 42] are effective in part because they engage students in strategic uses of testing.

While frequent low-stakes testing has been shown to be beneficial, it hinges on having access to large amounts of practice questions for students to work on—and writing (and refining) questions requires significant instructor effort. Adding to this burden, many faculty re-write assignments every semester to mitigate plagiarism. This cycle of constant question development can be mitigated by *question generators* [57, 58], questions that have been augmented with code to generate a range of question versions (see Figure 1). Question generators can be re-used across terms, because each student gets a fresh, randomized question from the generator. In addition, they make it easy to give students as much practice as they want by allowing them to keep generating new question versions.

Question generators solve a number of problems, but they do not solve the problem of authoring effort. In fact, creating and testing questions generators and the associated code takes even longer than writing static questions. However, question generators are well suited to sharing across courses and instructors, because their re-use is less detrimental to question security due to randomization [23].

Open educational resources (OER) are freely available and publicly shared resources that provide a platform for this infrastructure [59, 60]. While there are ample OER resources that provide

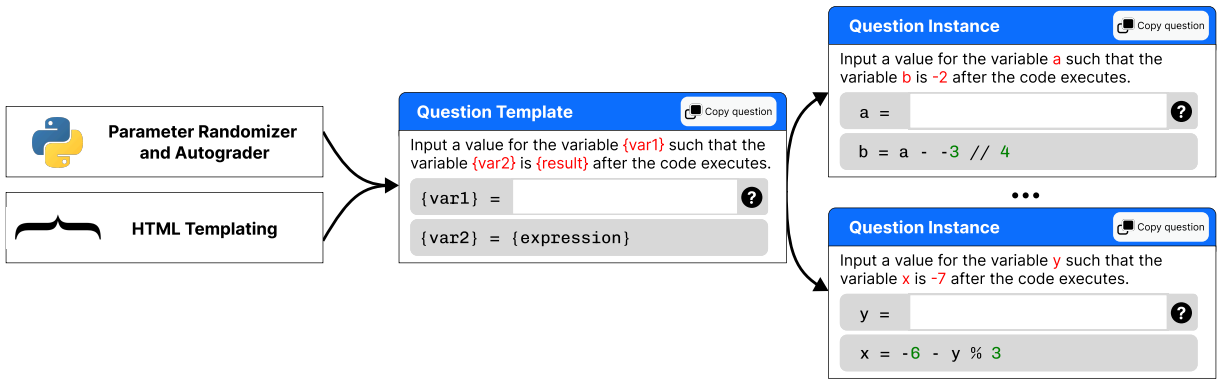

Figure 1: An example question generator with one-click copy question feature. The portions of the question marked in red (the target value and the arithmetic expression) are generated in a randomized manner by a snippet of Python code—they will appear differently to each student who attempts the question. This allows students to get more practice and instructors to worry less about cheating [23]. Randomized parameters need not be highlighted in a different color in the user interface.

content for students, we lack OER platforms that provide automatically gradable assessments [40, 60]. OER question generators can meet this critical need.

PrairieLearn is an open-source platform that provides infrastructure for authoring and sharing OER question generators. It is being used across 20 institutions. This paper introduces the PQSS, a system for sharing questions which is directly integrated into PrairieLearn. Integrating the question sharing system directly into the same software where instructors and students are using the questions in their courses has the potential to reduce the many documented barriers to the sharing of questions as OER [30, 40, 60].

While trying to share question generators in a form that could be supported in any platform would be ideal, this paper attempts a simpler goal, which still has yet to be solved: efficiently sharing questions among instructors within a single platform. We believe that sharing even at this level is sufficiently challenging in both technical and social dimensions to merit discussion and may inform future attempts to share within and across platforms.

2 Background

2.1 Open Educational Resources

It is widely accepted that in order for educational content to be considered OER, it must meet the 5Rs: users must be able to retain, reuse, revise, remix, or redistribute the educational content [40, 59, 60]. OER are "teaching, learning, and research resources that reside in the public domain or have been released under an intellectual property license that permits their free use or re-purposing by others" [18]. As such, content must be licensed in a way that has these features, with the most commonly chosen licenses being in the Creative Commons set of licenses [1, 40, 60].

A significant driver for the adoption of OER is to improve access to materials for students in developing countries and those who simply cannot afford the cost of traditional textbooks and accompanying materials [56]. Instructors also appreciate modifying OERs as opposed to starting from scratch.

Unfortunately, OER typically focus on presenting information and not providing assessments [40, 60]. Even prominent reports on the progress of OER fail to mention assessments at all [18]. While OER question banks exist (e.g., KEEN Cards [38], nanoHub [39], clark.center [2], Open Physics Problem Bank [7]), they are often isolated from other OER platforms and lack randomization and/or automated grading. Wiley asserts that open assessment is an area that "merit[s] particular attention" [60], and Knox agrees, noting that a key weakness of OER is the poor assumption that learners are able to self-assess [40].

2.2 Barriers to the adoption of OER

Despite the many benefits of using OER, many people are still hesitant to share their own educational resources, or to use OER. Some barriers to sharing and usage are:

(1) *Effort*: Instructors don't want to spend the time it takes to publish their materials given other time demands [30].
(2) *Security*: Many instructors are worried about losing the integrity of their exams if they publish their questions [30].
(3) *Recognition*: Instructors would like to receive credit for their published educational resources, especially for promotion at their universities [30].
(4) *Time Concerns:* It takes time to integrate resources into your course [21].
(5) *Lack of Quality:* Some faculty have concerns that OERs may not be of the same quality as traditional, proprietary resources [20].
(6) *Discoverability:* Many faculty said they had issues finding resources, worried that content for the courses they teach didn't exist, and wanted peer reviewed repositories that were easily searchable [15, 20].

In Section 3.2, we will discuss how PQSS was specifically designed to address many of these concerns, both on the side of question generator authors sharing their generators and on the side of those using shared question generators. We will also discuss plans in place to address additional barriers in future updates.

2.3 Question Interchange Formats

We briefly review two question interchange efforts: the Question and Test Interoperability specification (QTI) [14] and the efforts of

Table 1: Summary of the barriers to sharing and use of OER, and how PQSS is designed to resolve these concerns.

Barriers	Solution in PQSS	Status
Using OER:		
Time Concerns	Questions are directly importable from within PrairieLearn	**Complete**
Quality	PQSS will provide psychometrics on past usages of the questions	*Planned*
Discoverability	PQSS will provide recommendations based on similar content	*Planned*
Sharing OER:		
Effort	Questions can be directly shared within PrairieLearn	**Complete**
Security	Questions can optionally be shared privately	**Complete**
Recognition	PQSS will provide usage statistics	*Planned*

the Standards, Protocols, and Learning Infrastructure for Computing Education (SPLICE) group [52].

The goal of QTI is to provide a single format that multiple learning management systems can use for interoperable questions and exams [14]. QTI uses its own XML format with support for automatic grading, feedback via adaptive items [13], and item generators via item templates [12]. While QTI is supported by major LMS vendors, we have not seen substantial efforts to share questions via QTI in the SIGCSE community, possibly because (1) it uses a custom XML format to describe question randomization that is not easily generated from standard programming languages, and (2) it has no support for grading student code.

The SPLICE working groups have been developing standards and systems that facilitate sharing of assessments and other smart content that are CS specific [52]. One such standard is their Programming Exercise Markup Language (PEML) [24], which focuses on the ease of translation to other markups and an intuitive key/value pair structure. The PQSS system described in this paper isn't trying to solve the interoperability challenges that QTI and SPLICE are trying to solve, so it permits a much richer programming interface, but has the potential to simplify the process of sharing, as it can be done directly through the platform.

2.4 PrairieLearn

PrairieLearn is an open source online question and answer platform [57]. PrairieLearn empowers instructors to create sophisticated question generator types (e.g. programming, Parsons problems, symbolic mathematics, interactive student graphs, and autograded written responses) with HTML. Some examples are shown in Figure 2. PrairieLearn's robust plugin architecture allows instructors to extend the platform however they want by writing custom code: instructors can create, or integrate existing, interactive tools for students in the browser and develop custom grading scripts on the back end. The open source model for PrairieLearn enables faculty to contribute new question types to the community, facilitating the spread of PrairieLearn to new disciplines. PrairieLearn has been adopted by hundreds of courses at dozens of universities [10]. PrairieLearn's collection of features makes it a great host for an integrated question sharing system.

PrairieLearn fills a critical need in the current landscape of educational sharing and autograding systems. Most OER sharing systems (e.g., OpenStax [8], MIT Open-courseWare [6], edX [4], LibreText [5], CMU Open Learning Initiative [43]) are focused on delivering information rather than supporting assessment and are often closed ecosystems (e.g., they lack integration with Learning Management Systems (LMSs)) [40, 60]. Other sharing platforms are Discipline-Based Sharing Systems (DBSS; e.g., KEEN Cards [38], nanoHub [39], clark.center [2], Open Physics Problem Bank [7]) that encourage faculty to share assessments for a single discipline but generally do not provide autograding or randomization and are not easily extensible to new domains. Conversely, most autograding systems are either proprietary (Codio [3], WebAssign, Gradescope [53], Zybooks [32]) or are closed ecosystems (Runestone [27]). The lack of integration between sharing and grading systems means that instructors must often manually port content from a sharing system to an LMS, creating a major barrier to adoption of OER [30]. With the implementation of PQSS, PrairieLearn addresses these issues by uniquely allowing instructors to share questions and consume shared questions in the same platform that delivers the questions to students and autogrades them.

3 Implementation

While some implementation details of PQSS are specific to PrairieLearn, we believe the features and design principles of the system could be informative to other questions and answer platforms. Question generators can either be shared publicly, so that anyone can use them in their course, or shared only to specific other courses using the private sharing system. For clarity, we will refer to the course that is sharing a question generator as the "producing course" or "producer" and to the course using a shared question generator in their assessments as the "consuming course" or "consumer".

Because a question generator on PrairieLearn is created by code, each generator needs an entire execution environment. At minimum, this execution environment comprises an HTML file that defines the user interface, a Python file that generates and grades question variants, and a JSON configuration file with question metadata. However, a question generator may also rely on additional server or client code. Sharing this execution environment is cumbersome. However, PQSS handles this complexity automatically by executing the generator in the context of the producer rather than the consumer. Because shared question generators are executed in the same manner as they exist in the producing course, updates and bug fixes are automatically propagated to consumers, helping resolve concerns about OER maintenance.

Question generators are shared in PrairieLearn by editing the configuration files for the generators, which can be done directly in the JSON files or through the web user interface. Thus, sharing question generators does not require creating any new accounts or

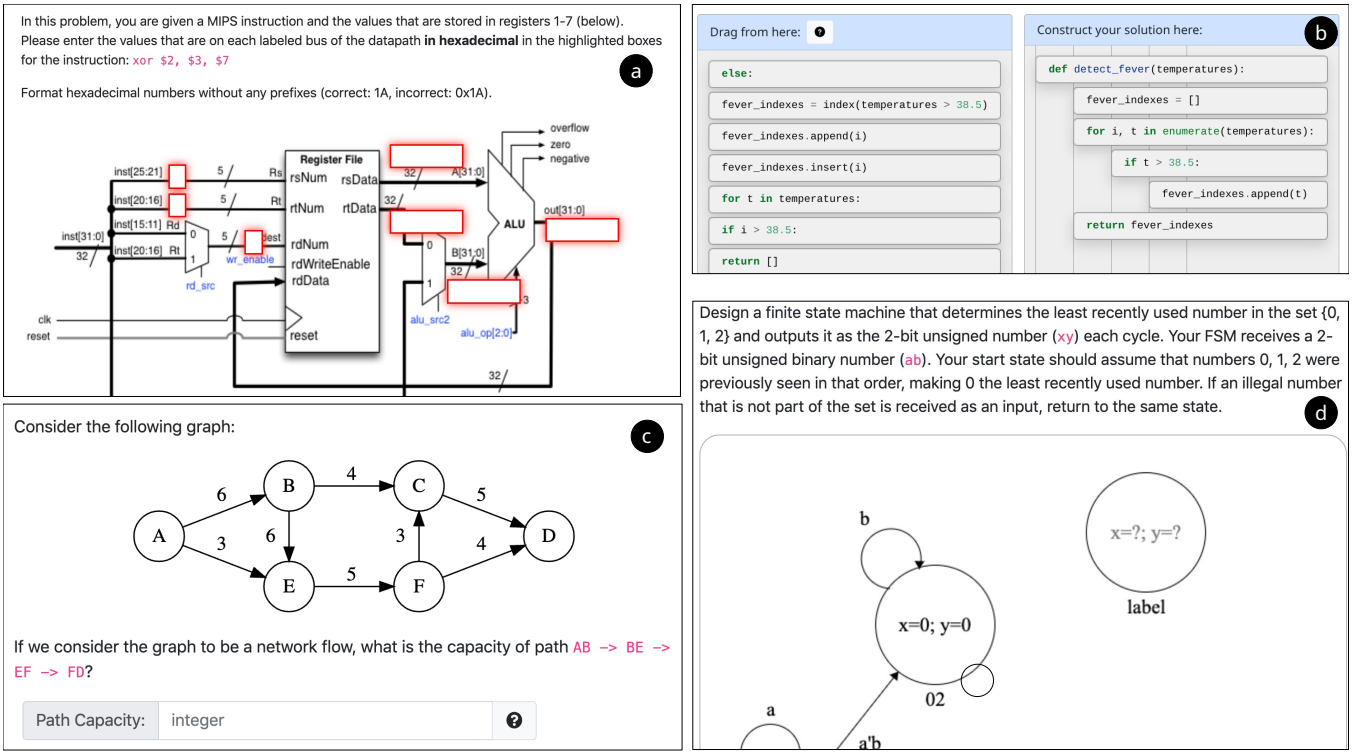

Figure 2: Examples showing flexibility of PrairieLearn: (a) label the datapath value for a randomly generated instruction, (b) Parson's problem, (c) randomly parameterized graph, and (d) an FSM drawing mini-CAD tool. PrairieLearn also supports many other types of questions, including grading student code in any language, autograded Explain-in-plain-English question using NLP, or any other type of problem for which an instructor can build a user interface with HTML and JavaScript, and can grade using Python code.

generator conversion, minimizing the amount of time it takes to share questions.

Assessments (homework and exams) in PrairieLearn consist of a list of which question generators are on the assessment (referred to by their question ID), along with other configuration details such as how many points each question should be worth, how many attempts students should get, and other information relevant to exam. These can be edited in the web user interface, or as JSON files. To refer to a question generator from another course, the system uses the question ID prefixed by the `"@"` symbol and the sharing name of the course. For example, to use the question generator `"addNumbers"` from the course with sharing name `"test-course"`, one would put `"@test-course/addNumbers"` into their assessment's JSON configuration.

3.1 Sharing Publicly

PQSS supports two types of public sharing: import sharing and source sharing. Both import and source sharing can be turned on and off for questions on an individual basis. Questions set for import sharing (with `"sharePublicly": true`) or for source sharing (with `"shareSourcePublicly": true`) are considered as being published for free use under a creative commons license license [1]. Question authors may choose to share their questions either one or both ways, depending on their needs. More details are shown in the PQSS documentation [9].

3.1.1 Import Sharing. Import sharing allows other people to directly use a question in their assessments. Import sharing is a good way to make it easy for other instructors to use questions written by other instructors or for other courses. Import sharing will also propagate all updates and fixes made to questions to the consuming course. Once a question has been shared and then imported by another course, that version of the question cannot be unshared to prevent breaking the consuming course.

On the other hand, using only import sharing provides some measure of security over the inner workings of a question generator because the source code is not publicly available, so it may not be clear how new variants of the question are generated.

3.1.2 Source Sharing. Source sharing allows other users to preview, view the source, and copy the source of a question. Instructors interested in making their content maximally available for others to use will likely enable both import sharing and source sharing of their questions.

There are some use cases, however, where an instructor may want to enable only source sharing without import sharing. For example, an instructor may want to avoid the potential maintenance burden of other instructors relying on their questions remaining

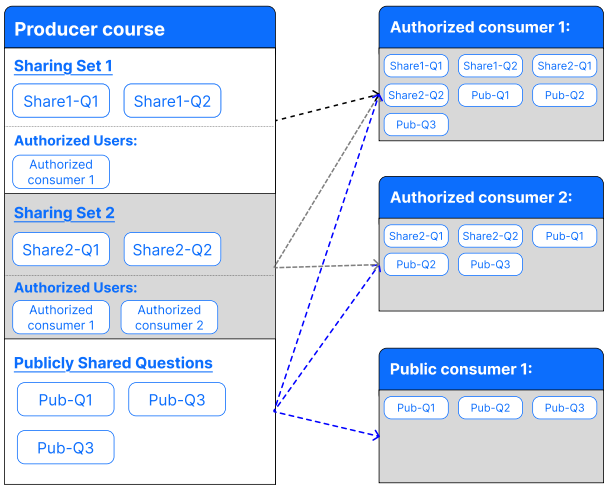

Figure 3: Overview of the question sharing system, showing the different levels of question-sharing permissions.

usable over time. Or, if a question has some context that makes it too course-specific, the question author might realize that consumers of the question will always want to copy and modify it before use rather than just importing it directly.

3.2 Private Sharing Through Sharing Sets

While many instructors want to share their course content publicly for anyone to use, there are also instructors that want to share only to a limited consumer set. We provide the sharing set mechanism to resolve instructors' worries about exam security in relation to sharing questions [30]. Access to shared questions that are not shared publicly is controlled through *sharing sets*. A sharing set is a named set of questions that you can share directly with another course. Course owners choose which courses or instructors can access which of their sharing sets. On the consumer end, importing is the same as through public import sharing. For security reasons, establishing the connection for one course to share questions with another course requires coordinated action by producers and consumers to approve the connection [9].

3.3 Sharing Assessments and Course Instances

Sharing an entire assessment or course instance is done by setting `"shareSourcePublicly": true` in the JSON configuration file for the assessment or course instance. This will enable anyone to view or copy the JSON configuration file for the assessment or course instance that is being shared. This will help those sharing questions to organize in a way that will make it even easier for question consumers to adopt, improving discoverability.

3.4 Authorship Credit and Usage Statistics

Those who write questions can specify their information in the JSON configuration for the question. This collection of authorship information will enable PrairieLearn to gather usage statistics for a given author's questions. The administrators of PrairieLearn can, at a user's request, generate a report with usage statistics for an author's questions. In the future, we plan to build a dashboard where question authors can view their question usage statistics automatically. This transparency of usage statistics resolves the concern that many instructors have, that they will not receive credit for their work [30], as these usage reports can demonstrate the impact of their work during tenure and promotion evaluation.

3.5 Additional Benefits of Sharing Integration

There are a few additional benefits of PQSS being directly integrated into PrairieLearn, which will help further address instructors' concerns with using OER. Because the sharing system is a part of the same software system where students are answering the questions and having them graded, we can provide anonymized aggregate statistics to give evidence of the quality of questions (such as their difficulty and discrimination parameters [36]). This can help resolve instructors concerns about quality of the OER questions.

We also plan to address instructors concerns about discoverability and content coverage by building a recommendation system that can recommend OER questions to instructors that are about similar content as the content they are covering in their own course. We wil also have a searchable catalog of questions that instructors can browse. These will also be enabled by the integration of PQSS into PrairieLearn.

4 Usage Experiences

PQSS has been used over the last few years by many instructors on our authorship team, as well as by other users of PrairieLearn. So far, 782 questions have been shared publicly and 571 questions have been shared through sharing sets by a total of 31 different courses on PrairieLearn. Here we highlight some of the experiences of the authors in using PQSS.

4.1 Enabling Concept Inventory Sharing

Concept inventories are validated assessments that are designed to evaluate students' knowledge. They can be used for research purposes, to test the efficacy of different teaching strategies, or as diagnostic tools for instructors to test students' misconceptions.

The Dynamic Programming Concept Inventory (DPCI) [29], Cybersecurity Concept Inventory (CCI) [47], and Cybersecurity Curriculum Assessment (CCA) [34] are all hosted on PrairieLearn. While many concept inventories for computer science have been created, many of them have not been used past their initial creation and validation [16, 47].

For the ongoing validation of the Dynamic Programming Concept Inventory, the question sharing system in PrairieLearn has been hugely beneficial as all of the instructors collecting data on the exam from their courses only have to import the questions, so there can be a single source of truth for the question text across revisions, without different versions of questions needing to be copied around to different locations [29].

4.2 Enabling Rapid Course Development

The first at-scale use of PQSS took place in sharing introductory programming questions from the University of Illinois to University of California San Diego, both large research institutions in the United States. The consuming course was a large introductory

Python course that had recently undergone a significant redesign to incorporate Generative AI into course instruction and incorporating PrairieLearn as the platform for administering all course assessments. Without the ability to share questions, this adoption would have been highly challenging. It would have required either manually copying all questions and their question elements from the producing course to the consuming course—a process prone to the errors and limitations detailed in Section 3.2—or authoring an entirely new question bank, an undertaking that would have been intractable in any reasonable period of time.

The question sharing mechanism ended up being essential to the success of the new course. Author Porter was the lead instructor of the course and faced the challenge of developing new learning goals, new lectures, new labs, and new assessments while also piloting PrairieLearn for the first time on campus. As PrairieLearn was new to the campus, neither the instructors nor the instructional assistants were familiar with the platform and how to author questions. As deadlines for the release of assignments loomed, the ability to draw on a large repository of existing questions freed up time to create new questions unique to this course and ultimately made the course possible (and successful).

Partially because this first use of sharing at scale happened before the implementation of shared assessments and courses, it did come with significant overhead. The question bank for the producing course was large which posed difficulties for the consumer to discover questions that were suitable for their context without the aid of the producer. Though this was easily remedied through weekly meetings, such meetings are, of course, only possible when a producer has the time and willingness to support the consumer in the question discovery process. In the future, the ability to share assessments and course instances as a whole will help instructors identify entire sets of questions for a given topic at once.

4.3 Enabling Rapid Prototyping and Evaluation

A central feature of the PrairieLearn platform is that it enables the creation of custom web components (known as "elements" in PrairieLearn). This makes it an ideal platform for the creation and use of customized user interfaces, assessment items, and autograding approaches, to name a few. However, prior to question sharing, if such items were to be used in multiple courses, this would require the custom elements to be copied to each course in which the element is used, introducing the risk of question version divergence. PQSS allows elements and questions to be authored and managed from a centralized repository, enabling the rapid creation and adoption of new tools.

This rapid prototyping may be particularly beneficial to educational research, empowering easy sharing across multiple universities. To the best of our knowledge, PQSS has been used to this effect at least three times. Two instances were for the evaluation of a novel LLM based autograding approach for code comprehension questions [54]. The autograder and custom question elements that utilized the autograder were managed from a central, producer repository. This enabled easy adoption of the autograder across multiple courses, and multiple institutions, with minimal overhead. The third instance was for pilot testing of a novel grading mechanism for Parsons Problems across multiple courses [46].

4.4 OER Course Repositories

As an initial prototype of assembling a public repository of OER question generators, we have solicited contributions from existing users of PrairieLearn. These are presented on a dedicated "OER resources page" on the PrairieLearn homepage [11]. There are currently eleven course repositories listed, with questions contributed from 16 distinct authors. Anecdotally, these question generators are being copied by other users of PrairieLearn, and they will serve as the basis of our upcoming recommender and search systems as described in Section 3.5. We are currently soliciting additional content for this collection and we anticipate that it will continue to scale up as PQSS becomes more widely used and additional features are added.

5 Conclusion and Future Work

The implementation in PQSS is a significant step forward in simplifying both the sharing of questions and using questions shared by other instructors. It can therefore address the critical need for more assessments in OER [40, 60]. By integrating question sharing features into an online question and answer platform, PrairieLearn, we seek to address many of the points of friction which have prevented people from sharing questions and using shared questions in the past. Future work on PQSS will continue to address more of the documented pain points of sharing questions by making it easier for authors to receive credit for their work and making it easier for question consumers to find relevant material for their courses. Our learning from implementing PQSS can also help build a body of knowledge to make it easier for other platforms to implement question sharing between instructors, as well as to work on future standards for interoperability across platforms.

References

[1] 2024. *About CC Licenses.* https://creativecommons.org/share-your-work/cclicenses/ Accessed: 6-27-2024.
[2] 2024. *clark.center.* https://clark.center/ Accessed: 6-27-2024.
[3] 2024. *Codio.* https://www.codio.com/ Accessed: 6-27-2024.
[4] 2024. *edX.* https://www.edx.org/ Accessed: 6-27-2024.
[5] 2024. *LibreTexts.* https://libretexts.org/ Accessed: 6-27-2024.
[6] 2024. *MIT OpenCourseware.* https://ocw.mit.edu/ Accessed: 6-27-2024.
[7] 2024. *Open Physics Problem Bank.* https://firas.moosvi.com/oer/physics_bank/files/about.html Accessed: 6-27-2024.
[8] 2024. *OpenStax.* http://openstax.com/ Accessed: 6-27-2024.
[9] 2025. PrairieLearn Content Sharing Documentation. https://prairielearn.readthedocs.io/en/latest/contentSharing Accessed: November 2025.
[10] 2025. PrairieLearn Marketing Page. https://www.prairielearn.com/ Accessed: 6-23-2025.
[11] 2025. PrairieLearn Open Educational Resources (OER): Questions. url https://www.prairielearn.com/catalog/oer. https://www.prairielearn.com/catalog/oer Accessed: 6-23-2025.
[12] 1EdTech. 2012. *1EdTech Question & Test Interoperability Assessment Test, Section and Item Information Model.* https://www.imsglobal.org/question/qtiv2p1/imsqti_infov2p1.html Accessed: June 2025.
[13] 1EdTech. 2022. *1EdTech Final Release Spec Version 3.0.* https://www.imsglobal.org/spec/qti/v3p0/terms Accessed: June 2025.
[14] 1EdTech. 2022. *QTI Specification.* https://www.imsglobal.org/spec/qti/v3p0/oview Accessed: 6-4-2025.
[15] IS Abeywardena, G Dhanarajan, and Chee Seng Chan. 2012. Searching and locating OER: Barriers to the wider adoption of OER for teaching in Asia. In *Proceedings from the Regional Symposium on Open Educational Resources: An Asian perspective on policy and practices.* 19–21.
[16] Murtaza Ali, Sourojit Ghosh, Prerna Rao, Raveena Dhegaskar, Sophia Jawort, Alix Medler, Mengqi Shi, and Sayamindu Dasgupta. 2023. Taking Stock of Concept Inventories in Computing Education: A Systematic Literature Review. In *Proceedings of the 2023 ACM Conference on International Computing Education Research -*

Volume 1 (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 397–415. https://doi.org/10.1145/3568813.3600120

[17] K. M. Arnold and K. B. McDermott. 2013. Free recall enhances subsequent learning. *Psychonomic bulletin & review* 20 (2013), 507–513. https://doi.org/10.3758/s13423-012-0370-3

[18] D. E Atkins, J. Seely Brown, and A. L Hammond. 2007. A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities. (2007). https://oerderves.org/wp_content/uploads/2007/03/a_review_of_the_open_educational_resources_oer_movement_final.pdf

[19] R. L. Bangert-Drowns, J. A. Kulik, and C.-L. C. Kulik. 1991. Effects of Frequent Classroom Testing. *The Journal of Educational Research* 85, 2 (1991), 89–99. https://doi.org/10.1080/00220671.1991.10702818

[20] Olga Maria Belikov and Robert Bodily. 2016. Incentives and barriers to OER adoption: A qualitative analysis of faculty perceptions. *Open praxis* 8, 3 (2016), 235–246.

[21] TJ Bliss, John Hilton III, David Wiley, and Kim Thanos. 2013. The cost and quality of open textbooks: Perceptions of community college faculty and students. (2013).

[22] B.S. Bloom. 1968. Learning for mastery. *Evaluation Comment* 1, 2 (1968), 1–12.

[23] B. Chen, M. West, and C. Zilles. 2018. How much randomization is needed to deter collaborative cheating on asynchronous exams?. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale (L@S '18)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3231644.3231664

[24] CS Splice. 2019. PEML. https://cssplice.org/peml/index.html publisher: CS Splice.

[25] J. Dunlosky, K. A. Rawson, E. J. Marsh, M. J. Nathan, and D. T. Willingham. 2013. Improving Students' Learning With Effective Learning Techniques: Promising Directions From Cognitive and Educational Psychology. *Psychological Science in the Public Interest* 14, 1 (2013), 4–58. https://doi.org/10.1177/1529100612453266 arXiv:https://doi.org/10.1177/1529100612453266 PMID: 26173288.

[26] D. S. Dustin. 1971. Some effects of exam frequency. *The Psychological Record* 21 (1971), 409–414.

[27] B. J. Ericson and B. N. Miller. 2020. Runestone: A platform for free, on-line, and interactive ebooks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 1012–1018.

[28] J. Feldman. 2023. *Grading for equity: What it is, why it matters, and how it can transform schools and classrooms.* Corwin Press.

[29] Matthew Ferland, Varun Nagaraj Rao, Arushi Arora, Drew van der Poel, Michael Luu, Randy Huynh, Freddy Reiber, Sandra Ossman, Seth Poulsen, and Michael Shindler. 2025. Construction and Preliminary Validation of a Dynamic Programming Concept Inventory. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSE 2025)*.

[30] Max Fowler, David H. Smith IV, Binglin Chen, and Craig Zilles. 2023. "I Don't Gamble To Make My Livelihood": Understanding the Incentives For, Needs Of, and Motivations Surrounding Open Educational Resources in Computing. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 430–443. https://doi.org/10.1145/3568813.3600136

[31] F. E Fulkerson and G. Martin. 1981. Effects of exam frequency on student performance, evaluations of instructor, and test anxiety. *Teaching of Psychology* 8, 2 (1981), 90–93. https://doi.org/10.1207/s15328023top0802_7

[32] C. Gordon, R. Lysecky, and F. Vahid. 2021. The shift from static college textbooks to customizable content: A case study at zybooks. In *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–7. https://doi.org/10.1109/FIE49875.2021.9637289

[33] G. L. Herman, Z. Cai, T. Bretl, C. Zilles, and M. West. 2020. Comparison of Grade Replacement and Weighted Averages for Second-Chance Exams. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (Virtual Event, New Zealand) *(ICER '20)*. Association for Computing Machinery, New York, NY, USA, 56–66. https://doi.org/10.1145/3372782.3406260

[34] Geoffrey L. Herman, Shan Huang, Peter A. Peterson, Linda Oliva, Enis Golaszewski, and Alan T. Sherman. 2023. Psychometric Evaluation of the Cybersecurity Curriculum Assessment. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 228–234. https://doi.org/10.1145/3545945.3569762

[35] G. L. Herman, K. Varghese, and C. Zilles. 2019. Second-chance testing course policies and student behavior. In *Proceedings of the 49th ASEE/IEEE Frontiers in Education Conference*. Cincinnati, OH, 7 pages. https://doi.org/10.1109/FIE43999.2019.9028490

[36] Natalie Jorion, Brian Gane, Katie James, Lianne Schroeder, Louis V. DiBello, and James Pellegrino. 2015. An Analytic Framework for Evaluating the Validity of Concept Inventory Claims. *Journal of Engineering Education* 104 (10 2015), 454–496. https://doi.org/10.1002/jee.20104

[37] J. D. Karpicke. 2009. Metacognitive control and strategy selection: Deciding to practice retrieval during learning. *Journal of Experimental Psychology: General* 138 (2009), 469–486. https://doi.org/10.1037/a0017341

[38] KEEN. 2024. *KEEN Cards*. https://engineeringunleashed.com/card Accessed: 6-27-2024.

[39] G. Klimeck, M. McLennan, S. P. Brophy, G. B. Adams III, and M. S. Lundstrom. 2008. nanohub.org: Advancing education and research in nanotechnology. *Computing in Science & Engineering* 10, 5 (2008), 17–23. https://doi.org/10.1109/MCSE.2008.120

[40] Jeremy Knox. 2013. Five critiques of the open educational resources movement. *Teaching in Higher Education* 18, 8 (Nov. 2013), 821–832. https://doi.org/10.1080/13562517.2013.774354 Publisher: Routledge _eprint: https://doi.org/10.1080/13562517.2013.774354.

[41] N. Kornell, M. J. Hays, and R. A. Bjork. 2009. Unsuccessful retrieval attempts enhance subsequent learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 35 (2009), 989–998. https://doi.org/10.1037/a0015729

[42] Chen-Lin C. Kulik and James A. Kulik. 1987. Mastery Testing and Student Learning: A Meta-Analysis. *Journal of Educational Technology Systems* 15, 3 (1987), 325–345. https://doi.org/10.2190/FG7X-7Q9V-JX8M-RDJP arXiv:https://doi.org/10.2190/FG7X-7Q9V-JX8M-RDJP

[43] Farhat (Meena) Lakhavani and Mary Ann Blair. 2011. Using the Open Learning Initiative (OLI) to Deliver Digital Citizenship Instruction to University Freshmen at Carnegie Mellon University. In *Proceedings of EdMedia + Innovate Learning 2011*, Theo Bastiaens and Martin Ebner (Eds.). Association for the Advancement of Computing in Education (AACE), Lisbon, Portugal, 2744–2749. https://www.learntechlib.org/p/38248

[44] R. R. Martin and K. Srikameswaran. 1974. Correlation between frequent testing and student performance. *Journal of Chemical Education* 51, 7 (1974), 485. https://doi.org/10.1021/ed051p485

[45] E. Mazur. 1997. *Peer Instruction: A User's Manual*. Prentice Hall, Upper Saddly River, NJ.

[46] Skyler Oakeson, David H. Smith IV, Jaxton Winder, and Seth Poulsen. 2025. Choose Your Own Solution: Supporting Optional Blocks in Block Ordering Problems. arXiv:2510.11999 [cs.HC] https://arxiv.org/abs/2510.11999

[47] Seth Poulsen, Geoffrey L. Herman, Peter A. H. Peterson, Enis Golaszewski, Akshita Gorti, Linda Oliva, Travis Scheponik, and Alan T. Sherman. 2021. Psychometric Evaluation of the Cybersecurity Concept Inventory. *ACM Trans. Comput. Educ.* 22, 1, Article 6 (Oct. 2021), 18 pages. https://doi.org/10.1145/3451346

[48] K. A. Rawson, J. Dunlosky, and S. M. Sciartelli. 2013. The power of successive relearning: Improving performance on course exams and long-term retention. *Educational Psychology Review* 25 (2013), 523–548. https://doi.org/10.1007/s10648-013-9240-4

[49] H. L. Roediger, III and J.D. Karpicke. 2006. The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science* 1 (2006), 181–210. https://doi.org/10.1111/j.1745-6916.2006.00012.x

[50] H. L. Roediger, III and J.D. Karpicke. 2006. Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science* 17 (2006), 249–255. https://doi.org/10.1111/j.1467-9280.2006.01693.x

[51] H. L. Roediger, III and J.D. Karpicke. 2008. The critical importance of retrieval for learning. *Science* 319 (2008), 966. https://doi.org/10.1126/science.1152408

[52] Cliff Shaffer, Peter Brusilovsky, Ken Koedinger, Thomas Price, Tiffany Barnes, and Behrooz Mostafavi. 2024. Ninth SPLICE Workshop on Technology and Data Infrastructure for CS Education Research. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. 1904–1904.

[53] A. Singh, S. Karayev, K. Gutowski, and P. Abbeel. 2017. Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In *Proceedings of the fourth (2017) ACM conference on Learning@ Scale*. 81–88. https://doi.org/10.1145/3051457.3051466

[54] David H Smith IV and Craig Zilles. 2024. Code Generation Based Grading: Evaluating an Auto-grading Mechanism for" Explain-in-Plain-English" Questions. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. 171–177.

[55] B. R. S. Sutopo, M. Nurtanto, S. Purnomo, N. A. Handoyono, and A. B. Johan. 2020. Enhancing of student involvement and collaboration through think-pair-share model on energy conversion learning. *International Journal of Higher Education* 9 (2020), 199–205. https://doi.org/10.5430/ijhe.v9n4p199

[56] UNESCO. 2002. Forum on the Impact of Open Courseware for Higher Education in Developing Countries. 3 (2002).

[57] M. West and G. L. Herman. 2015. PrairieLearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In *Proceedings of the 2015 American Society for Engineering Education Annual Conference and Exposition*. Seattle, WA.

[58] M. West, M. S. Sohn, and G. L. Herman. 2015. Randomized exams for large STEM courses spread via Communities of Practice. In *Proceedings of the 2015 American Society for Engineering Education Annual Conference and Exposition*. Seattle, WA. https://doi.org/10.18260/p.24575

[59] D. Wiley. 2010. Openness as catalyst for an educational reformation. In *Educause Review*. Educause. https://openaccess.uoc.edu/bitstream/10609/7201/1/Wiley_TheOpenFuture_2010_eng.pdf

[60] D. Wiley, T. J. Bliss, and M. McEwen. 2014. Open Educational Resources: A Review of the Literature. In *Handbook of Research on Educational Communications and Technology*, J. M. Spector, M. D. Merrill, J. Elen, and M. J. Bishop (Eds.). Springer, New York, NY, 781–789. https://doi.org/10.1007/978-1-4614-3185-5_63