education sciences

MDPI

Article

A Real-Time Collaborative Learning Platform Integrated with a Document Management System

Linda Burchiellaro [1,*], Francesco Faenza [2], Cecilia Marchetti [2], Mariane Joseph Anzeda [2], Martina Casari [2] and Claudia Canali [2]

[1] Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, 41125 Modena, Italy
[2] Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, 41125 Modena, Italy; francesco.faenza@unimore.it (F.F.); cecilia.marchetti@unimore.it (C.M.); marianejoseph.anzeda@unimore.it (M.J.A.); martina.casari@unimore.it (M.C.); claudia.canali@unimore.it (C.C.)
* Correspondence: linda.burchiellaro@unimore.it

Abstract

The declining interest in STEM disciplines and the limitations of existing cloud-based collaborative tools, including restricted real-time collaboration, privacy concerns, and limited institutional integration, highlight the need for open-source, on-premises platforms that combine notebook-based computation with document management workflows, support synchronous editing, and preserve institutional control over data, access policies, and software environments without proprietary cloud services. This paper presents NextPyter, an open-source platform integrating Jupyter Notebooks with Nextcloud. Originally designed for research, it was adapted for educational use to enable real-time co-editing of computational notebooks within a familiar, file-centric environment. The platform underwent an exploratory evaluation through two university-level programming activities involving first- and second-year Computer Engineering students. The evaluation used the Perceived Usefulness (PU) construct of the Technology Acceptance Model (TAM), complemented by reflexive thematic analysis of open-ended responses. Quantitative results indicate high internal consistency of the PU scale in both cohorts (Cronbach's $\alpha = 0.94$ and 0.86) and mean PU scores of 4.09 and 3.53, respectively, on a five-point scale. All participants considered the platform useful for learning. Qualitative findings highlight positive perceptions of peer interaction, collaborative coding, and experiential learning. These exploratory findings suggest that NextPyter may support collaborative notebook-based learning while preserving institutional data governance.

Keywords: collaborative learning; interdisciplinary education; open-source platform; jupyter notebooks; nextcloud

Check for updates

Academic Editors: Maria Beatrice Ligorio and Chris Campbell

Received: 26 January 2026
Revised: 17 October 2026
Accepted: 2 November 2026
Published: 6 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In recent years, several European reports and international studies by Deloitte (2025) and European Commission and Directorate-General for Education, Youth, Sport and Culture (2025) have highlighted a steady decline in students' interest in STEM disciplines (Science, Technology, Engineering, Mathematics) among school-aged learners. This phenomenon represents a significant challenge for educational systems, which are increasingly called upon to identify tools and methodologies that stimulate curiosity, support active learning,

and foster the acquisition of scientific and technological skills increasingly demanded by the labor market.

A key factor underlying this disengagement is the persistence of traditional, transmissive teaching methods, which limit active participation and reduce students' perception of the relevance of acquired knowledge. Research consistently advocates for collaborative, inquiry-based learning strategies as more effective alternatives (Attard et al., 2021; Ješková et al., 2022). These approaches require educational environments that support peer interaction, shared workspaces, and iterative experimentation.

Existing digital tools, including Google Colaboratory, Kaggle, Deepnote, and Replit, offer advanced programming capabilities in cloud environments but fall short in educational contexts: they often lack reliable real-time collaboration, raise privacy and data governance concerns, and impose usage constraints in their free tiers that limit institutional adoption. There is, therefore, a growing need for platforms that combine ease of use, real-time collaboration, and institutional control, while remaining accessible to both students and educators without requiring advanced technical skills.

NextPyter, the platform presented in this paper, addresses these challenges by integrating open-source technologies such as Jupyter Notebook (Jupyter Notebook, 2026), Nextcloud (NextCloud, 2026), and Docker (Docker, 2026) to enable the sharing of educational resources, real-time collaborative editing, and joint work on interactive notebooks. This integration results in a dynamic, participatory, and easily accessible environment that does not require advanced technical skills, making it suitable for use in secondary and higher education contexts.

The origins of the NextPyter platform lie in a different but closely related domain. The platform was originally conceived as a tool to support interdisciplinary collaborative research and was first presented in Faenza et al. (2024), developed with the primary goal of facilitating cooperation among researchers through shared computational environments and collaborative workflows. Building on the positive outcomes of this initial research-oriented deployment, NextPyter has evolved into an open-source platform explicitly tailored for innovative and collaborative education, and this evolution itself constitutes one of the primary contributions presented in this paper. This evolution introduced new collaborative and educational features, as well as a redesign of the user experience to better support classroom activities and group-based experimentation. In its current form, NextPyter is designed to be tested and evaluated through real classroom experiences, allowing for a systematic assessment of its students' perceptions of its usefulness for supporting learning, collaboration, and student engagement.

This work makes two complementary contributions. First, from a technological perspective, it presents NextPyter, an open-source, self-hosted platform that integrates Jupyter Notebooks and Nextcloud to support real-time collaborative programming within institutionally managed environments. Second, from an educational perspective, it reports an exploratory classroom evaluation investigating how students perceive the usefulness of this integrated environment during collaborative programming activities. By combining the description of the platform with evidence collected in authentic teaching settings, the study examines not only the feasibility of deploying NextPyter in university courses, but also the educational activities for which students perceive it as valuable. The evaluation focuses on students' perceived usefulness and qualitative feedback, rather than on directly measured learning outcomes.

Specifically, the current version of NextPyter introduces the following set of innovative features that distinguish it from conventional educational and research platforms:

- Open-source foundation: promoting transparency, flexibility, and adaptability to different educational and research contexts.

- Interdisciplinary focus: fostering collaboration among students, educators, and researchers from different fields, thus encouraging the exchange of knowledge and methodologies.
- Real-time collaboration: enabling multiple users to work simultaneously on shared notebooks and resources, improving coordination and productivity.
- Ease of use: designed to be accessible even to users without advanced technical skills, thereby broadening participation and reducing technological barriers.
- Integration with institutional infrastructures: ensuring seamless interoperability with existing systems such as authentication services and learning management platforms.

The platform has been tested in an educational context through two university-level computer science activities, where students' perceived usefulness of the platform was assessed using the Perceived Usefulness (PU) construct of the Technology Acceptance Model (TAM), complemented by qualitative analysis of student feedback. Results indicate a generally positive perception, highlighting NextPyter's potential to support collaborative programming and bridge the gap between the need for and the actual use of digital tools in education. Given the relatively small and homogeneous sample ($n = 29$), drawn from two classroom cohorts within a single university-level computer science program, this study is intended as an initial exploratory classroom evaluation rather than as evidence supporting generalizable conclusions across different disciplines or institutional contexts. The findings should therefore be interpreted as preliminary evidence of the platform's usability and potential, while broader conclusions would require further evaluation with larger and more diverse samples across multiple educational settings.

The release of NextPyter as a fully open-source platform, publicly available at (Faenza et al., 2026), represents a concrete contribution of this work, allowing institutions and developers to freely deploy, inspect, and extend the system to meet their specific educational and organizational requirements.

The remainder of this paper is structured as follows. Section 2 reviews the motivation and background, covering STEM enrollment trends, pedagogical frameworks, and emerging educational needs, collaborative digital platforms, and existing digital tools for teaching and learning. Section 3 describes the NextPyter platform, detailing its deployment model, design features for collaborative classroom use, and integration with institutional infrastructures. Section 4 presents the evaluation methodology, including the questionnaire design based on the Perceived Usefulness construct of the TAM model, the use case scenarios, and the qualitative analysis approach. Section 5 reports and discusses both the quantitative and qualitative results, together with the platform's strengths and limitations. Section 6 concludes with final remarks and future directions.

2. Motivation and Background

This section examines the context motivating the development of NextPyter along four complementary dimensions. We first analyze current trends in STEM enrollment to establish the broader educational challenge (Section 2.1). Then, we discuss the pedagogical frameworks and practical needs that guide the platform's design choices (Section 2.2). In Section 2.3, we review collaborative digital platforms, integrated environments that combine multiple services to support coordinated learning or research workflows, distinguishing them from standalone digital tools, which are the subject of the final Section 2.4. While platforms provide unified ecosystems with shared workspaces, authentication, and document management, tools typically offer specific functionalities, such as notebook execution or code editing, without broader integration into institutional infrastructures. This distinction is central to understanding the gap that NextPyter addresses: the absence of an open-source,

on-premises platform that unifies collaborative notebook execution with institutional file management within a single, accessible environment.

2.1. Student Enrollment in STEM

STEM competencies are widely recognized as essential for sustaining Europe's competitiveness, technological leadership, and capacity to address the green and digital transitions. According to the Education and Training Monitor 2025 (European Commission and Directorate-General for Education, Youth, Sport and Culture, 2025), the EU faces persistent shortages of STEM specialists, particularly in engineering and ICT, despite these fields offering some of the highest employment rates for recent graduates. Projections to 2035 show sustained growth in STEM occupations, but the supply of qualified graduates remains inadequate, with the EU trailing the UK and Canada and ranking last in ICT graduate output. Against this backdrop, student participation in STEM pathways has remained largely stagnant over the past decade: the 2024 Deloitte report (Deloitte, 2025) reveals that only 26.6% of European tertiary students are enrolled in STEM programs, with women representing just 31.9%. Participation in ICT is even more limited, with only 19.5% of STEM students selecting this specialization and women comprising merely 20%. Although numerous national and EU-level initiatives have been launched to increase STEM participation and address the gender gap, including targeted strategies and programmes explicitly aimed at women (European Commission and Directorate-General for Education, Youth, Sport and Culture, 2025; Faenza et al., 2021a, 2021b).

A key factor underlying this trend is the persistence of traditional, lecture-based teaching approaches that fail to stimulate curiosity and engagement, particularly among younger learners. The Rocard Report (Rocard et al., 2007), published by the European Commission in 2007, was among the first to document this problem systematically and to advocate for more active, inquiry-based pedagogies as a structural response. Subsequent policy documents, including the 2015 report "Science Education for Responsible Citizenship" (European Commission and Directorate-General for Research and Innovation, 2015) and OECD-PISA data (OECD Publishing, Paris, 2023), have reinforced this diagnosis, calling for curriculum revision and the adoption of educational environments that support active participation and hands-on experimentation. The growing importance of informal and digital learning, with 44% of STEM students relying on online and non-traditional training (Deloitte, 2025), further underscores the need for accessible, technology-enhanced educational platforms capable of bridging formal instruction and self-directed practice.

2.2. Emerging Needs in Education and Innovative Didactics

One of the main factors contributing to students' disengagement from scientific subjects is the persistence of traditional, transmissive teaching methods, centered on frontal lectures and decontextualized content that limit active participation and reduce students' perception of the relevance of acquired knowledge (Sælemyr & Bjørndal, 2019). Two well-established pedagogical frameworks offer a structured response to these limitations and directly inform the design rationale of NextPyter.

Collaborative Learning defines learning as a fundamentally social process in which learners construct knowledge through structured peer interaction and mutual negotiation (Dillenbourg, 1999), a principle rooted in Vygotsky's sociocultural theory and the concept of the Zone of Proximal Development (Vygotsky, 1978). In technology-enhanced contexts, Computer-Supported Collaborative Learning (CSCL) extends this framework by highlighting how appropriately designed digital environments can scaffold shared problem-solving and collective knowledge construction (Stahl et al., 2006). Complementarily, Inquiry-Based Learning (IBL) positions student-driven exploration, hypothesis formulation, and

iterative experimentation as the primary mechanisms of knowledge acquisition (Pedaste et al., 2015), and has been explicitly endorsed for STEM education by the Rocard Report (Rocard et al., 2007) and subsequent European Commission policy (European Commission and Directorate-General for Research and Innovation, 2015). Together, these frameworks converge on a common set of requirements for educational technology, support for active participation, peer interaction, shared workspaces, and iterative experimentation, all of which directly inform the design of NextPyter and are increasingly recognized as central to effective STEM education (Legarde et al., 2025; Xinh et al., 2025).

Within computer science education specifically, several challenges have been identified. The most prominent concern is the lack of tools that can be easily adopted by teachers and students without advanced technical skills. Traditional development environments are often difficult to configure and maintain, especially for novices who lack programming experience (Kelter et al., 2019). Another limitation involves file management and code sharing, particularly in collaborative projects. For curricular courses, integrating source code and instructional materials into a unified environment would offer substantial benefits, helping students focus on core concepts instead of the technical complexities of configuring and managing tools (Cristiano & Triana, 2019).

These limitations are also reflected in recent classroom-based studies on collaborative platforms. While systems such as CollaClassroom and RiPPLE have shown positive results in improving engagement and learning outcomes in controlled or course-specific settings, their evaluations highlight challenges related to scalability, integration into existing teaching practices, and accessibility for users with varying technical skills (Khosravi et al., 2019; Sayeed et al., 2025).

These needs, combined with the increasing use of programming languages with a low entry barrier, such as Python (Grandell et al., 2006; Weigend, 2020), have encouraged the adoption of solutions like Jupyter Notebooks for organizing computer science learning activities (Kluyver et al., 2016). These environments support exploratory learning, immediate feedback, and hands-on experimentation, closely aligning with principles of active learning and constructivist pedagogy (Campbell et al., 2024; Tufino et al., 2024; Vallejo et al., 2022). Empirical evidence supports this combination: Mannila et al. (2006) studied 60 novice programmers aged 16–19 and showed that students using Python made significantly fewer syntax and logic errors than those using Java, and that those who later transitioned from Python to Java experienced no disadvantages, reinforcing the suitability of Python-based notebook environments for introductory programming education.

Previous studies have shown that integrated platforms such as Google Classroom facilitate collaborative work, support learning environments, and document sharing by combining editing, storage, and communication tools within a unified environment (Nardito et al., 2025). However, such solutions rely on proprietary infrastructures that limit institutional control over data storage and access.

Given the need for a flexible, on-premises, open-source alternative, NextPyter adopts Nextcloud as its file management backbone, giving institutions full control over data storage and access, thereby strengthening privacy protections and supporting compliance with educational data regulations. The platform offers an intuitive, user-friendly interface that enables both teachers and students, regardless of their technical background, to fully exploit its functionalities. As an open-source and modular platform, NextPyter can furthermore be extended through integrations with other widely used services such as Google Drive, adapting to diverse institutional contexts without sacrificing data governance.

2.3. Collaborative Digital Platforms for Educational and Research

Since the previous section highlighted the limitations of existing proprietary solutions and motivated the need for flexible, institution-controlled infrastructures, it is natural to broaden the perspective and examine how such requirements are addressed at the level of full-fledged platforms. In particular, rather than focusing on isolated functionalities, recent work has increasingly explored collaborative digital platforms as integrated environments capable of supporting complete educational and research workflows. These platforms aim to unify multiple services, such as content management, computation, user modeling, and collaboration, within a single ecosystem, aligning with the same principles of integration, extensibility, and data governance that underpin the design of NextPyter.

For instance, CollaClassroom proposes an AI-augmented collaborative learning environment integrating large language models to support both individual and group study activities. The platform was evaluated through a small-group study with 12 university students in Bangladesh. The evaluation used a questionnaire combining closed-ended items on collaborative learning, AI/LLM usage, usability, and perceived learning impact with open-ended questions about participants' experiences and perceptions of AI integration. Some items were adapted from established frameworks, including the TAM and usability heuristics. The findings indicated high usability, low frustration, and strong receptivity to LLM-assisted collaboration, with participants reporting improved engagement and conceptual understanding (Sayeed et al., 2025). CollaClassroom is designed for institutional deployment (on-premises infrastructure), though it does not integrate a native document management system for file sharing and storage beyond the learning environment itself. Similarly, RiPPLE introduces a crowdsourced adaptive learning ecosystem in which students and instructors collaboratively generate educational content, while recommendation algorithms personalize learning pathways based on learner modeling techniques. Its evaluation involved a pilot study in an introductory relational databases course at The University of Queensland, with 453 students. The analysis examined measurable learning gains, students' ability to identify effective learning resources, and their perceptions of the platform's contribution to learning. Students were divided into groups according to their engagement with RiPPLE, allowing the authors to compare learning outcomes between participants who actively used the platform and those who did not. The results suggested measurable learning gains and positive perceptions of RiPPLE's educational value (Khosravi et al., 2019). Like CollaClassroom, it operates within institutional boundaries (on-premises infrastructure), but it lacks integrated document management capabilities. In the context of computational STEM education, OSSCAR presents an open, web-based platform that leverages Jupyter notebooks and interactive widgets to deliver modular educational applications accessible through standard web browsers (Du et al., 2022). OSSCAR focuses on the design and functionality of the platform and illustrates its potential applications in education, while providing limited empirical evidence on student learning outcomes or technology acceptance. While these systems demonstrate the pedagogical value of AI support, adaptive recommendation, and interactive notebook-based delivery, they primarily focus on platform-level functionalities, such as content personalization, workflow orchestration, or large-scale resource integration. Although some existing platforms offer real-time collaboration or on-premises deployment individually, comparatively less attention has been devoted to solutions that simultaneously combine all three elements (real-time collaborative notebook editing, on-premises institutional infrastructure, and native document management) within a single framework, particularly in contexts that bridge research collaboration and didactic activities.

Against this background, the present study provides exploratory evidence from the use of NextPyter in authentic university classroom activities, focusing on students' perceived

usefulness and complementing quantitative ratings with qualitative feedback on its potential educational applications. This evaluation offers initial insights into how students perceive the value of an integrated computational workspace for collaborative programming and practice-oriented learning, while leaving its impact on learning outcomes and collaborative processes for future investigation.

Parallel efforts in the research domain have focused on supporting workflows through notebook-based environments. Notebook-as-a-VRE (NaaVRE) extends JupyterLab into a collaborative virtual research environment where notebook cells can be containerized into reusable components and organized into workflows (Zhao et al., 2022). While it supports on-premises deployment, it does not offer real-time collaborative notebook editing or native document management. Similarly, the Agave Platform provides a Science-as-a-Service solution with APIs for job management, data handling, and application deployment across hybrid infrastructures (Dooley et al., 2018). Unlike the platforms discussed above, Agave functions as middleware for scientific job orchestration rather than as a collaborative learning environment; it lacks real-time notebook co-editing and integrated document management. Large-scale facilities such as the European Spallation Source have integrated JupyterHub with Docker to deliver scalable and secure computing environments (Fernández et al., 2016). Similarly, this deployment targets scalable computational access for research workflows rather than classroom collaboration, and does not provide the real-time editing or document management features required for the educational platform comparison. Likewise, SciTokens (Withers et al., 2018), which enables fine-grained delegable access to distributed scientific data, is an authentication and authorization mechanism rather than a platform; as such, it provides neither real-time editing nor document management capabilities.

Despite their strengths in scientific computing, workflow management, and secure access, these platforms are primarily designed for specialized research infrastructures rather than everyday educational use. Consequently, they do not fully address the need for environments that integrate notebook-based computation with familiar collaborative document management tools. This gap motivates the development of integrated platforms that combine containerized notebooks with institutional file sharing systems, supporting both collaborative learning and research within a unified and accessible framework.

Within this landscape, NextPyter is proposed as a platform for educational settings that addresses many of the limitations of existing solutions while providing an integrated ecosystem for teaching, learning, and, more broadly, collaboration. By combining open-source components, such as Nextcloud for collaborative file management and Jupyter Notebooks for interactive computation, the platform enables students to focus on programming and problem-solving without being constrained by local setup and configuration overhead.

Moreover, the open-source, modular design of NextPyter (Section 3) supports institutional deployment requirements, particularly regarding data governance and privacy, and enables adopters to extend the platform through additional integrations and backend execution options. Compared to mainstream cloud notebook services, NextPyter's core differentiator is the combination of self-hosting and a user-friendly interface that enables real-time co-editing of notebooks while preserving familiar file sharing and permission semantics.

2.4. Digital Tools for Teaching and Learning

In contrast to full-fledged platforms, digital tools typically provide standalone functionalities designed to support specific teaching or learning activities (e.g., coding, visualization, or content sharing), without offering a fully integrated environment for managing the entire educational or research workflow.

In recent years, numerous tools have emerged to support collaborative and interactive teaching, fostering active learning and student participation. Kaggle (Kaggle, 2026) is widely used in the data science community for sharing datasets and notebooks, but it lacks real-time collaboration and is not designed for structured educational use. Limitations such as session time constraints, limited environment customization, and the absence of student monitoring tools further restrict its applicability in classrooms. Google Colaboratory (Colab) (Bisong, 2019) provides accessible cloud-based execution of Jupyter Notebooks with free computational resources. However, it offers limited computational resources, no real-time collaboration, and raises concerns related to privacy and institutional integration, particularly due to its proprietary nature. Deepnote (Deepnote, 2026) supports collaborative notebooks with advanced features such as simultaneous editing and class management, but its free tier is resource-limited, and its proprietary ecosystem may present challenges for privacy-sensitive educational environments. Moreover, it is offered as a cloud service without on-premises deployment options. Observable (Observable, 2026) enables collaborative data exploration and visualization, yet its free plan imposes significant restrictions, and data privacy concerns limit its use in academic settings. Binder (Project Jupyter, 2026) allows execution of notebooks from Git repositories without installation, but requires technical expertise for configuration, reducing its accessibility for less experienced users. Although Binder instances can be hosted on institutional servers (on-premises infrastructure), the tool provides no native document management capabilities. Replit (Replit, 2026) offers an online development environment with collaborative and AI-assisted features, but its free version limits the number of active projects, constraining scalability in educational contexts.

While these tools provide valuable features for designing engaging educational activities, they generally operate in isolation and lack the level of integration, extensibility, and institutional control offered by platform-based solutions. They also present important limitations: free versions often include restricted functionalities, while full access typically requires paid licenses that may not be sustainable for many educational institutions. Additionally, limited flexibility, scalability, and concerns related to privacy and data management can further hinder their adoption in structured teaching contexts.

To highlight the positioning of NextPyter with respect to existing solutions, Table 1 summarizes the platforms and tools discussed above along three key dimensions: real-time collaboration (simultaneous multi-user editing), on-premises infrastructure (deployment within institutional boundaries), and native document management (integrated file sharing, storage, and permission semantics).

Table 1. Comparison of collaborative educational platforms.

Platform and Tool	Real-Time Collaboration	On-Premises Infrastructure	Native Document Management
CollaClassroom	✓	✓	×
RiPPLE	✓	✓	×
OSSCAR	×	✓	×
NaaVRE	×	✓	×
Kaggle	×	×	×
Google Colaboratory	×	×	✓
Deepnote	✓	×	✓
Observable	✓	×	×
Binder	×	✓	×
Replit	✓	×	×
NextPyter	✓	✓	✓

Note: ✓ indicates that the feature is supported, whereas × indicates that it is not supported.

3. NextPyter Platform

Collaborative notebook-based activities in programming courses can be hindered by operational barriers. In fact, in many learning contexts, the main obstacles are not the notebook itself, but rather the surrounding complexity: installation, authentication, environment configuration, file exchange, and support for synchronous collaboration. As seen in Section 2, many existing notebook platforms lack seamless integration with document management or impose constraints on self-hosted deployments.

NextPyter addresses these gaps by embedding Jupyter notebooks within Nextcloud's familiar file-sharing environment. As a fully open-source platform, it can be deployed on-premises, allowing institutions to manage data and access policies locally. Because the entire stack is open-source, institutions can freely inspect, modify, and extend the platform to meet their specific needs while retaining full control over data and access policies. This approach promotes the transparency, flexibility, and adaptability that educational contexts require.

From the adoption perspective, deployment flexibility is important because adoption in educational institutions is rarely uniform. Some organizations can experiment with a fully prepackaged installation, while others already have identity-management systems, storage services, or Nextcloud instances in place. The present work formalizes the platform architecture and deployment model, focusing on the goal of reducing deployment complexity for both adopters and contributors.

This section outlines the key design decisions and provides a condensed technical overview of the underlying architecture.

3.1. Deployment Model and Institutional Integration

NextPyter is distributed as a modular stack that supports both full deployment and partial integration into existing infrastructures. The core components are packaged separately: the front-end integration is distributed as a standard Nextcloud application, while the back-end execution layer is distributed as a set of container images. In addition, a `docker-compose` configuration is provided to automatically instantiate the full reference stack, including an authentication layer based on Keycloak, a Nextcloud instance with the NextPyter app pre-installed, the daemon responsible for spawning, routing, and garbage collection of notebook servers across different containerization systems, and an Nginx reverse proxy for request routing. This setup lowers the technical threshold for pilot deployments, while still allowing more mature institutions to reuse only the components they need.

Figure 1 provides a layered view of the NextPyter architecture and clarifies the separation between the end-platform layer and the container-spawning layer. This division, in addition to providing modularity, also provides abstraction for the end user.

3.2. Interdisciplinary Focus

NextPyter was designed for educational settings in which students and instructors may have heterogeneous technical backgrounds and limited time for software setup and troubleshooting. In these contexts, the practical barriers to notebook-based work often lie not in the notebook paradigm itself, but in the surrounding requirements: environment configuration, file exchange, account management, and the coordination of collaborative work. For this reason, the platform was designed around four classroom-oriented requirements: low technical friction, support for synchronous collaboration, continuity between notebooks and shared course materials, and compatibility with institutional infrastructures.

Rather than treating notebooks as isolated technical artifacts, NextPyter embeds them in a familiar shared-workspace model. This allows users to interact with computational

notebooks through the same file-oriented logic already used for other learning materials, thereby reducing the need to learn separate workflows for storage, sharing, and execution. The goal is not simply to make notebooks available in the browser, but to make them easier to integrate into ordinary teaching and collaborative learning practices.

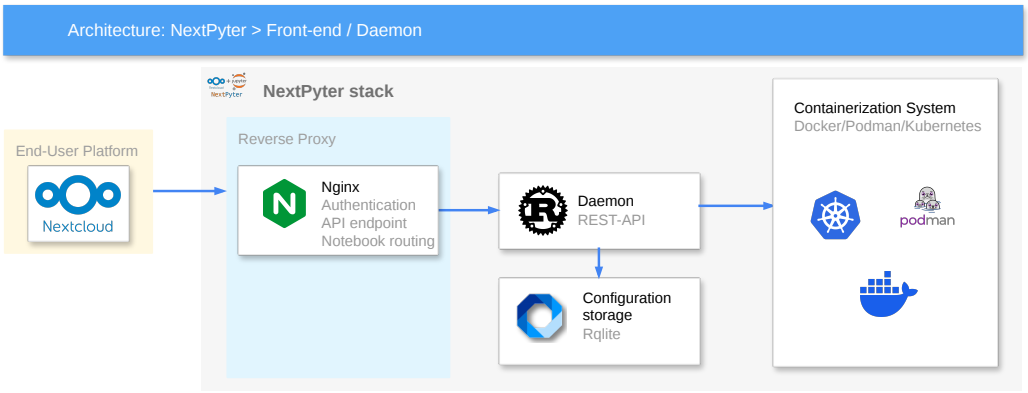


Figure 1. NextPyter: High-Level Architecture.

3.3. Real-Time Collaboration

A central requirement in classroom and project-based learning is the ability to work jointly on the same material without fragmenting the activity across multiple local copies or incompatible environments. NextPyter addresses this by extending the shared-folder model to notebooks, so that collaborators working within the same shared workspace can access the same notebook session and edit it synchronously. This supports group-based activities such as pair programming, guided lab work, collaborative exploration, and instructor-supported exercises, where students benefit from seeing changes immediately and working within a common computational context.

At the implementation level, real-time synchronization is enabled through the `jupyter-collaboration` extension and routed through the platform's existing HTTPS-based communication layer.

3.4. Ease of Use

NextPyter organizes collaboration around shared folders rather than individual notebook files. This choice reflects the way educational activities are usually structured in practice: a lab session, project, or assignment typically involves not only one notebook, but also datasets, intermediate outputs, reference material, and supporting documents. Associating the execution environment with the folder, therefore, provides a coherent workspace in which all related resources remain grouped under the same access and sharing logic.

This design also reduces configuration effort. All notebooks within the same folder hierarchy inherit the same execution image, meaning students do not need to configure each notebook separately, and instructors can prepare a consistent software environment for the entire activity. In educational terms, this helps maintain continuity across materials, reduces setup errors, and makes group work easier to coordinate, particularly when multiple users need access to the same resources.

3.5. Integration with Institutional Infrastructure

NextPyter is designed to fit within existing institutional ecosystems rather than replace them. By building on Nextcloud's storage and sharing model and supporting integration with established authentication services, the platform aligns with common practices for access control, synchronization, and document management. This reduces fragmentation for students and instructors, who can work within familiar institutional workflows, while the

separation between the collaboration layer and the execution back end preserves flexibility for future integrations.

4. Evaluation Methodology

This section describes the methodology adopted to evaluate NextPyter in real classroom settings. The evaluation was designed as an initial exploratory classroom study, focused on gathering preliminary evidence on the platform's perceived usefulness for collaborative programming activities and capturing students' qualitative feedback about their collaborative experience, rather than producing generalizable conclusions across different educational contexts. Given the sample and its focus on a single university-level computer science program, the methodological approach should therefore be understood as a context-specific assessment of the platform, providing a basis for subsequent evaluations involving larger and more diverse populations.

We first discuss the rationale for selecting the PU construct of the TAM as the primary evaluation measure (Section 4.1) and compare it with alternative instruments (Section 4.2). We then detail the questionnaire design (Section 4.3), describe the use case scenarios in which the platform was deployed (Section 4.4), and outline the qualitative analysis approach used to complement the quantitative findings (Section 4.5).

4.1. Evaluation Rationale and Model

In contemporary societies, where digital technologies permeate almost every aspect of daily and professional life, the utility of digital tools has become a key factor in determining their adoption, impact, and diffusion. In educational contexts, in particular, the quality of interaction between learners and technological systems directly influences motivation, learning outcomes, and the overall success of technology-mediated teaching. In educational contexts, it is important to evaluate the quality of interaction between learners and technologies to understand how technologies influence motivation, learning outcomes, and overall teaching.

To perform such an evaluation, the TAM model has been widely adopted as an empirically grounded method for its balance between academic rigor and functional simplicity: its structure consists of a concise number of constructs, which eases the submission process; moreover, TAM enables a clear, interpretable analysis of users' acceptance of digital tools, with particular attention to perceived benefits and adoption prospects, while avoiding the complexity of more extensive models such as UTAUT. TAM was originally introduced by Davis (1989) and is grounded in the Theory of Reasoned Action (Fishbein & Ajzen, 1975). It provides a well-established framework for explaining and predicting users' adoption of technology by identifying two core constructs: Perceived Usefulness (PU), defined as the degree to which a person believes that using a system increases task performance, and Perceived Ease of Use (PEOU), which reflects the perceived effort required to use the system.

Consistently with the exploratory scope of this classroom evaluation, the study adopts a TAM-based assessment focused exclusively on the PU construct, rather than evaluating the full TAM. This choice is supported by previous research identifying PU as the strongest and most stable predictor of acceptance intentions across both voluntary and mandatory usage contexts (Venkatesh & Davis, 2000), while also suggesting that PEOU judgments frequently overlap with PU in guided, instructor-led classroom settings (King & He, 2006). Accordingly, the questionnaire was restricted to the PU construct. This decision reduces respondent burden in a single-session design, avoids construct redundancy, and targets the study's primary objective: determining whether students perceive the platform as beneficial

for collaborative programming activities. The questionnaire included all six items of the original PU scale proposed by Davis (Davis, 1989).

As is standard in TAM-based evaluations, the questionnaire items in this study employ a Likert-scale format, and internal consistency is assessed using Cronbach's alpha.

4.2. Alternative Evaluation Models

Several standardized questionnaires and theoretical models have been proposed in the literature to assess effectiveness, user experience, and technology acceptance, each with different scopes, strengths, and limitations.

The User Experience Questionnaire (UEQ) (Schrepp et al., 2017) is a structured, widely validated instrument for assessing user experience across both functional and emotional dimensions, comprising 26 semantic differential items rated on a 7-point scale. It captures multiple aspects of UX, including attractiveness, efficiency, stimulation, and novelty, and is particularly effective for rapid, high-level evaluations in diverse contexts. However, UEQ focuses primarily on subjective experience rather than on technology adoption mechanisms. The partial overlap between dimensions and limited alignment with constructs such as Perceived Usefulness and intention to use reduce its suitability for studying acceptance processes in educational contexts, where understanding motivation and adoption is central.

The System Usability Scale (SUS) (Brooke, 1996) is a well-established, lightweight tool for obtaining a global measure of perceived system usability through ten Likert-scale items, producing a single usability score. Its simplicity and robustness make it effective for benchmarking and quick evaluations across a wide range of systems. Yet, SUS offers limited explanatory power, as it does not distinguish between different factors influencing acceptance, nor does it account for users' attitudes or intentions. Its generic nature makes it less suitable for educational studies.

The Usefulness, Satisfaction, and Ease (USE) questionnaire (Lund, 2001) evaluates usability through the dimensions of usefulness, ease of use, and satisfaction, aligning conceptually with some TAM constructs, particularly Perceived Usefulness and ease of use. Nevertheless, the questionnaire's length and descriptive focus limit its effectiveness for modeling acceptance dynamics. The USE questionnaire does not explicitly address behavioral intention or attitudinal mediation, both of which are essential for explaining why users adopt or continue using a specific technology.

UTAUT questionnaire (Venkatesh et al., 2003) provides a comprehensive and empirically powerful framework for predicting technology acceptance by integrating multiple theoretical models and incorporating social and organizational factors. Despite its strong predictive capabilities, UTAUT is less suitable than TAM for the educational context considered in this study due to the complexity and length of its standard questionnaire, which increases respondent burden. In contrast, TAM offers a more parsimonious structure that balances explanatory power with practical feasibility.

4.3. Questionnaire Design

The evaluation instrument was designed as a short, structured questionnaire combining closed-ended Likert-scale items with open-ended questions. This format was chosen to balance quantitative comparability with the richness of qualitative feedback, while keeping respondent burden low given the single-session classroom context.

The questionnaire focused exclusively on the Perceived Usefulness construct of the TAM, while Perceived Ease of Use was intentionally excluded. This choice was motivated by both methodological and contextual considerations. First, in guided classroom activities such as those described in this study, PEOU-related items tend to overlap with PU, as ease of use is often implicitly reflected in users' judgments of usefulness (King & He, 2006).

Second, as this study represents an initial evaluation of NextPyter in a real classroom setting, the primary objective was to assess whether the platform is perceived as meaningful and beneficial for learning activities. Focusing on PU therefore allowed us to avoid redundancy among constructs, keep the questionnaire concise, and reduce participants' cognitive load while still obtaining actionable insights to guide future design iterations.

To enrich the quantitative PU data, the study also incorporated qualitative methods through open-ended questions. Research has shown that structured questionnaires alone may overlook important insights, such as concerns about technology replacing human interaction or feelings of isolation (Felber et al., 2024). Integrating qualitative feedback enables participants to offer more detailed observations and suggestions, providing a complementary and deeper understanding of perceived usefulness.

The questionnaire was organized into three sections. The first section collected demographic information and investigated students' prior experience with collaborative tools in classroom settings, as well as their perceived need for collaborative platforms to support in-class activities. The second section focused on the PU construct of TAM, measured on a 5-point Likert scale from 1 (unlikely) to 5 (likely). This section also included specific items assessing students' perceived utility of NextPyter during the lesson and their willingness to use the platform in future classroom activities. The third section consisted of two open-ended questions aimed at gathering qualitative feedback, allowing participants to suggest potential improvements to the platform and to provide free-form observations based on their experience.

The complete questionnaire is provided in the Appendix A, allowing readers to examine all items in detail and facilitating transparency and reproducibility of the study.

4.4. Use Case

The classroom activities were designed to explore NextPyter as a shared environment for collaborative programming exercises. From an educational perspective, the activities aimed to give students opportunities to jointly develop solutions, discuss implementation choices, and receive support from instructors and tutors in a shared computational workspace. The evaluation focused on how students perceived the usefulness of this environment after participating in the activities, rather than on measuring improvements in programming skills or learning outcomes.

This section presents two real-world usage scenarios involving first-year and second-year cohorts, selected to observe how the platform's design choices, integration with Nextcloud, container-based isolation, and real-time collaboration meet the needs of users with different levels of prior experience in authentic classroom settings.

To evaluate NextPyter in a technical course, the platform was used during the Fundamentals of Programming course for first-year Computer Engineering students. The same experimental setup was replicated with second-year students enrolled in an advanced programming course, allowing us to observe the platform's applicability across different levels of technical proficiency. The activity aimed to assess whether enhanced interaction could be achieved during a standard university class. In particular, the goal was to enable rapid interaction among students and allow both the teacher and tutors to intervene directly in each group's work. The assigned activity involved modeling a simple software system using basic object-oriented principles for the first-year cohort, while the second-year students were tasked with a similar task but with a higher level of difficulty. Students were randomly assigned to different groups and asked to complete the task collaboratively using NextPyter. All students were given access to Nextcloud through the university's SSO system and assigned to a specific shared folder containing base materials, an initial notebook with the software design task, and additional resources. As described in the

architecture section, the platform allowed them to share notebooks, divide responsibilities, and edit content collaboratively in real time. At the same time, the professor and tutors were able to access each group's NextPyter folder, enabling them to monitor all group work and intervene directly on issues, offer suggestions, propose alternative implementations, provide explanations, and so on. A common repository for all students was also created and used by the professor to share important points that emerged during class, such as code samples and explanations that were provided to one group but considered useful for everyone. By working within a shared environment that integrates file management via Nextcloud and executable notebooks via Jupyter, students were able not only to collaborate via Jupyter notebooks but also to use other Nextcloud features, such as chat and file upload, to discuss design choices and jointly refine their solutions. The platform supported both the technical aspects of the task and coordination within each group, making collaboration an integral part of the learning experience. As will be shown in the results section, these aspects were among the most appreciated by students across both cohorts. Moreover, some of them explicitly asked at the end of the lesson whether this approach would be adopted more in the future, demonstrating the activity's success.

4.5. Qualitative Analysis

To complement the quantitative findings derived from the PU construct of the TAM, a qualitative analysis was undertaken to explore students' perceptions in greater depth. In particular, the open-ended responses provide insight into how participants interpret and envisage the use of the platform within authentic learning contexts. open The qualitative data were analyzed using reflexive thematic analysis, following Braun and Clarke's framework (Braun & Clarke, 2006). Reflexive thematic analysis was selected as it is particularly suitable for identifying recurring patterns in short open-ended responses and for supporting mixed-methods research designs. The approach was primarily inductive, allowing themes to emerge from the data, while remaining conceptually informed by the PU construct of the TAM in collaborative environments.

The data were first reviewed and lightly pre-processed to ensure consistency. An open coding phase was then conducted, during which descriptive labels were assigned to the responses. The coding process involved multiple researchers, each independently reviewing the data and proposing initial codes. These were subsequently compared and discussed within the research team in order to develop a shared coding scheme. Through this iterative process, codes were refined and grouped into broader categories, such as collaboration, project-based use, learning support, real-time interaction, and feature requests. The final coding structure was organized into a codebook including code definitions and representative examples. Given the exploratory nature of this component and the relatively small dataset, coding was conducted by four researchers working collaboratively, with emphasis placed on transparency, traceability, and inter-rater consistency throughout the analytical process.

5. Results and Discussion

We first present the quantitative findings derived from the questionnaire assessing the PU construct of the TAM, including item-level and composite Perceived Usefulness scores (Section 5.1), followed by the qualitative analysis of open-ended responses (Section 5.2). We then discuss the platform's strengths and limitations (Section Strengths of the Platform and Section 5.3) in light of both the empirical evidence and the design goals outlined in Section 3.

5.1. Quantitative Results

The questionnaire, as explained in the use case Section 4.4, was administered to two student cohorts after practical teaching sessions. The first administration took place within a Fundamentals of Programming course, delivered during the first semester of the first year of a Computer Engineering degree program. The second involved second-year students enrolled in a more advanced programming course. As described in Section 4.4, the lessons focused on Python programming and proposed exercises on core Object-Oriented Programming (OOP) concepts for both cohorts, with the complexity adjusted to match each cohort's level. During each session, students were randomly assigned to groups to collaboratively solve structured exercises using the NextPyter platform. Each activity lasted approximately two hours, allowing participants to become familiar with the platform and actively engage with collaboration features. Upon completion of the exercises, students were asked to complete the questionnaire assessing the PU construct of TAM, along with the associated open-ended questions.

The questionnaire was administered anonymously and was not associated with grading. Nevertheless, because responses were collected immediately after the activity, the results may reflect novelty effects and short-term impressions; they should therefore be interpreted as exploratory perceptions rather than stable evaluations. A total of 16 first-year students and 13 s-year students completed the questionnaire. In terms of demographics, the first-year cohort consisted of 14 male and 2 female students, while the second-year cohort included 10 male and 3 female students.

Regarding prior background, both cohorts presented a heterogeneous profile. Among first-year students, some had prior experience from technical secondary schools, while others had no previous exposure to programming. In the second-year cohort, a similarly mixed background was observed; however, all students had completed a first-year course in Fundamentals of Programming with Python, ensuring a more uniform baseline of programming knowledge.

The first part of the questionnaire asked whether students perceived a need to use collaborative tools and whether they had already used such tools for coding. Among the first-year cohort, most respondents (12 out of 16) reported a need for collaborative tools. In contrast, no one reported using any collaborative coding instrument. Regarding the second-year cohort, 7 out of 13 reported a need for collaborative tools, and 3 indicated regular use of Google Colaboratory. These differences align with the expected progression in programming exposure: first-year students are still forming their workflows, whereas second-year students have begun adopting tools independently.

5.1.1. Overall Perceived Usefulness

All respondents in both cohorts answered "Yes" to the question *"Do you think NextPyter could be useful for learning at your school/university?"*, indicating unanimous recognition of the platform's potential.

All six items of the original PU scale proposed by Davis (Davis, 1989) within the TAM demonstrated high internal consistency in both cohorts (first-year: Cronbach's $\alpha = 0.94$; second-year: Cronbach's $\alpha = 0.86$), supporting the use of a composite PU score calculated as the mean of the six items. Figure 2 displays the item-level distributions.

For first-year students, the composite PU score averaged 4.09 (SD = 0.71, Mdn = 3.92, IQR = 3.58–4.88). Second-year students reported a somewhat lower but still positive average of 3.53 (SD = 0.72, Mdn = 3.50, IQR = 3.33–3.83). The response distributions were concentrated in the upper half of the Likert scale for both groups, though first-year students showed consistently stronger agreement across most items. Figure 3 provides a combined

view of the composite scores, using boxplots with jittered individual points to illustrate the central tendency, spread, and individual variability within each cohort.

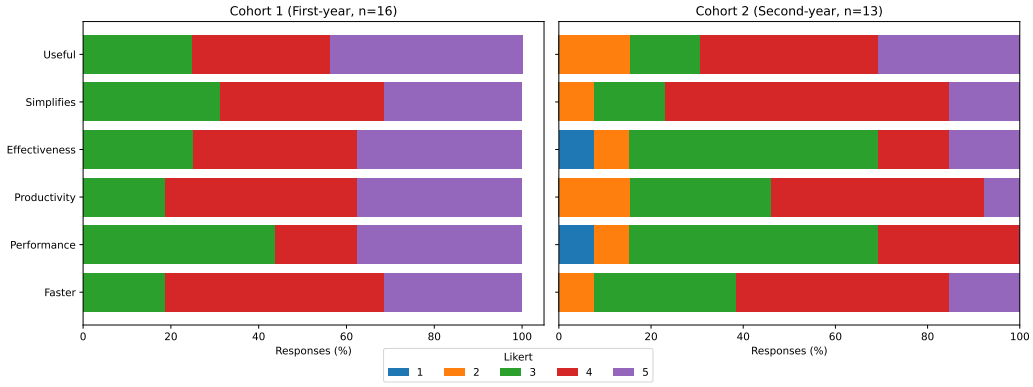


Figure 2. Perceived Usefulness (PU) item distributions.

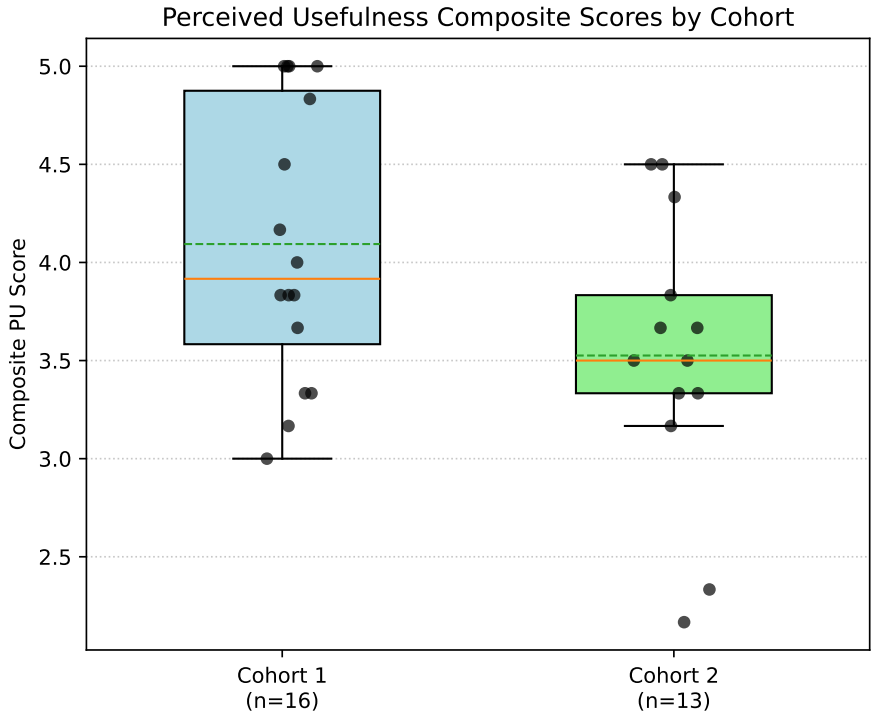


Figure 3. Composite PU score. Boxes show the interquartile range with the median (solid line) and mean (dashed line); whiskers extend to 1.5 × IQR; points show individual composite scores.

To assess whether the observed difference in composite scores was statistically meaningful, a Mann–Whitney U test was conducted (Shapiro–Wilk tests did not indicate severe departures from normality, but non-parametric methods were retained as a conservative choice given the small sample sizes). The test revealed a trend toward higher PU scores in the first-year cohort ($U = 146.0$, $p = 0.067$), with a small effect size (Cliff's $\delta = 0.20$). While not reaching conventional significance, this pattern is consistent with the different positions of the two cohorts in their university trajectory. The first-year cohort had just entered university, and none of its members had previously used any collaborative coding tool, despite most (12 of 16) reporting a need for one; for these students, NextPyter represents the first structured collaborative programming environment they encounter, and the gap it fills is correspondingly large. The second-year cohort, by contrast, had already completed a full year of coursework, including the Fundamentals of Programming course attended

by the first cohort, and had begun to adopt tools independently (three reported regular use of Google Colaboratory). Their lower, though still positive, PU scores therefore likely reflect a more mature and critical stance, as students with consolidated study habits and prior tool exposure weigh the incremental benefit of an additional platform against their established workflows. This interpretation is consistent with the TAM literature, where multi-study and longitudinal evidence shows that usefulness and acceptance judgments are sensitive to users' prior experience, evolving as working habits mature (Venkatesh & Bala, 2008; Venkatesh & Davis, 2000).

5.1.2. Item-Level Comparisons

Looking at individual PU items, first-year students rated all six dimensions higher on average than their second-year counterparts. The largest differences appeared in "Improve academic performance" ($\Delta M = 0.86$), "Increase learning effectiveness" ($\Delta M = 0.89$), and "Increase productivity" ($\Delta M = 0.73$). To control for multiple comparisons, a Bonferroni correction was applied (adjusted $\alpha = 0.05/6 = 0.0083$). Under this conservative threshold, none of the item-wise differences reached statistical significance. However, before correction, three items—"Improve academic performance" ($p = 0.034$), "Increase productivity" ($p = 0.034$), and "Increase learning effectiveness" ($p = 0.022$)—showed nominal significance, hinting at areas where the platform's impact may be felt most strongly by less experienced students.

Notably, all three items with the largest gaps are performance-oriented. For first-year students, who are still forming their first programming workflow, the potential productivity gains of a guided collaborative environment are highly salient; second-year students, possessing a more uniform baseline of programming knowledge, are more likely to evaluate such claims against their own experience, which yields more moderate expectations.

For second-year students, the highest-rated items were "Simplify study tasks" (M = 3.85) and "Useful for study path" (M = 3.85). The lowest rating was assigned to "Improve academic performance" (M = 3.08), possibly reflecting established study habits and a more critical stance toward the incremental benefits of new tools. Nonetheless, the positive evaluation of the platform's utility for their overall study path indicates that even experienced students see NextPyter as a worthwhile addition to their learning environment.

Overall, the composite PU scores observed in both cohorts (4.09 and 3.53 on a 5-point scale) reflect a clearly positive reception of the platform in both groups. These ratings are consistent with the broader TAM evidence, in which perceived usefulness routinely emerges as the strongest attitudinal driver of technology acceptance in educational settings (King & He, 2006; Scherer & Teo, 2019), while the antecedents of usefulness and acceptance judgments are known to shift as user experience grows (Venkatesh & Bala, 2008; Venkatesh & Davis, 2000).

5.1.3. Exploratory Sub-Group Analysis

Within the second-year cohort, students who reported prior use of collaborative coding tools ($n = 3$) had a median composite PU score of 3.33, compared to 3.67 for those without prior use ($n = 10$). A Mann–Whitney U test found no significant difference between these subgroups (U = 8.0, $p = 0.27$). The extremely small number of prior-tool users limits any firm conclusions, but the absence of a clear advantage for experienced users suggests that NextPyter's design may appeal broadly rather than only to novices.

5.2. Qualitative Results

The qualitative dataset was derived from a total of $n = 29$ participants (16 first-year and 13 s-year Computer Engineering students). From these participants, $n = 38$ textual units

were extracted and analyzed, including responses from the "Usage intention" ($n = 29$), "Suggestions" ($n = 6$), and "Additional comments" ($n = 3$) fields. Empty or non-informative responses were excluded from the analysis. Each response was treated as a unit of analysis.

The analysis revealed a set of recurring themes that collectively outline how students perceive the platform's role in supporting collaborative and practice-oriented learning. These themes represent interconnected dimensions of a broader perception of usefulness. The most prominent theme concerns the collaborative affordances of the platform. A large proportion of participants explicitly referred to group work and cooperation as primary use cases, using expressions such as "working in groups" and "to collaborate". These responses indicate that students primarily perceive the platform as a shared workspace that supports peer interaction and group-based learning activities. A second theme relates to project-based and practical use. Several participants indicated that the platform would be used for "group projects in class", highlighting its relevance for hands-on and applied learning scenarios. A third theme concerns learning enhancement and skill development. Some responses explicitly mention the potential of the platform to "to improve my practical knowledge", indicating that students perceive it as a tool that supports experiential learning and the development of technical competencies. A fourth theme relates to real-time coordination and remote collaboration. Although often implicit, several responses point to the importance of working together synchronously, suggesting that real-time interaction is perceived as a valuable feature of the platform. With regard to potential refinements, a smaller set of responses includes suggestions for improvement. These mainly concern the addition of communication features, such as "a separate comments section", indicating a need for more structured interaction tools within the platform. Table 2 summarizes the main themes, their descriptions, and representative excerpts, while Table 3 provides an overview of the coding scheme adopted in the analysis.

Table 2. Summary of qualitative themes, descriptions, and representative excerpts.

Theme	Description	Representative Excerpts
Collaborative affordances	Use for group work and peer interaction	"To collaborate"; "Working in groups"
Project-based learning	Use in practical and group-based activities	"Group projects in class"
Learning enhancement	Support for skill development	"To improve my practical knowledge"
Real-time collaboration	Synchronous and remote interaction	Implicit references across responses
Platform improvement	Suggestions for additional features	

Table 3. Example of coding scheme used in the qualitative analysis.

Code	Description	Example Quote
Collaboration	References to group work and peer interaction	"Working in groups"
Professional use	Use in practical or project activities	"Group projects"
Learning enhancement	Improvement of skills and knowledge	"Improving practical knowledge"
Real-time interaction	References to synchronous or remote collaboration	"Real-time collaboration"

Overall, the qualitative responses reveal a predominantly positive perception of the platform, with limited critical feedback. While this may reflect genuine satisfaction, it should also be interpreted cautiously, as open-ended questions tend to elicit shorter and less critical responses.

The qualitative findings not only support but also deepen the interpretation of the quantitative results. While the analysis of Perceived Usefulness captures students' perceptions at a structural level, the qualitative data reveal how this usefulness is concretely experienced by students, particularly through collaboration, shared problem-solving, and engagement in project-based activities. The strong emphasis on collaborative practices suggests that Perceived Usefulness is closely linked to the platform's ability to facilitate peer interaction and group-based learning. This aligns with existing research on collaborative educational environments (Attard et al., 2021; Ješková et al., 2022), which highlights the importance of integrating technical tools with social and interactive dimensions of learning. Furthermore, the association with project-based activities indicates that students perceive the platform as suitable for authentic and applied learning contexts, reinforcing its pedagogical value. In this sense, Perceived Usefulness emerges not merely as an individual evaluation, but as a relational and context-dependent construct. At the same time, the suggestions related to communication features highlight an important aspect: collaboration is not limited to shared editing but also involves structured interaction, feedback, and discussion.

These observations provide useful indications for both future platform development and the design of collaborative programming activities. More broadly, the classroom evaluation suggests that students' perceived usefulness of NextPyter is associated not only with its technical functionalities, but also with the opportunities for collaborative and practice-oriented activities that it offers. While these findings do not establish improvements in learning outcomes or the effectiveness of collaborative processes, they provide preliminary educational insights into how an integrated computational environment is perceived by students in authentic classroom settings.

5.2.1. Methodological Considerations

The qualitative component of this study is exploratory in nature and is based on a relatively small dataset of open-ended responses. While the analysis provides useful insights into students' perceptions, it is not intended to support statistical generalization.

The coding process involved four researchers who independently reviewed the responses and proposed initial codes. These were subsequently compared and discussed within the research team in order to refine the coding structure and agree on the final themes. This collaborative approach helped strengthen the consistency and robustness of the analysis, although no formal inter-coder agreement metrics were calculated.

Future research could further extend the qualitative component by examining larger samples and by collecting richer forms of qualitative data, such as semi-structured interviews, focus groups, or longitudinal feedback collected over time.

Strengths of the Platform

One of the most impactful educational scenarios for using NextPyter is a collaborative laboratory activity. Students organized into small groups can work simultaneously on the same Jupyter Notebook, shared through the integrated Nextcloud environment. Unlike several tools discussed in Section 2.4, where real-time collaboration is either limited, restricted in free versions, or constrained by proprietary models, NextPyter provides full collaborative functionality within an institutional deployment. Coding, textual documentation, and visualization coexist in a single digital workspace, enabling a holistic learning process without requiring external services or fragmented toolchains. Teachers can assign

programming exercises or other assigned tasks while monitoring group contributions in real time. Compared with the platforms analyzed in Section 2.3, where real-time collaborative notebooks and structured document management are typically not the primary focus within the same ecosystem, NextPyter unifies these dimensions in a single on-premises infrastructure. This design ensures institutional data governance and compliance while still enabling flexible access from school networks and remote locations. The result is a participatory and adaptable environment that fosters teamwork, peer learning, and the development of both technical and collaborative problem-solving skills.

A further advantage is the ability to provide immediate, real-time feedback. In contrast to platforms where feedback is asynchronous or dependent on separate communication channels, instructors can directly intervene within the shared Notebook, inserting comments, correcting code, or clarifying misconceptions as students work. This immediate feedback mechanism is designed to facilitate timely clarification and support students as they refine their solutions during collaborative activities. Its actual effects on student engagement and conceptual understanding, however, remain to be assessed through dedicated educational evaluations. Additionally, NextPyter enables systematic tracking and archiving of the full history of Notebook modifications through integrated synchronization and version control provided by the Nextcloud platform.

By combining collaborative notebooks, structured document management, institutional control, and real-time interaction within a deployable on-premises architecture, NextPyter addresses several limitations identified in current tools and platforms, offering a technically robust and pedagogically coherent solution for STEM education.

5.3. Limitations of Platform and Evaluation

Despite the advantages of integrating notebook execution into a document-oriented collaboration environment, the current version of NextPyter has limitations that affect both adoption and day-to-day use. At present, NextPyter is designed around Nextcloud as the end-user platform. While the daemon exposes a REST interface that could, in principle, be consumed by alternative clients, the mature user workflow (file browsing, sharing, and in-browser opening of notebooks) currently relies on the Nextcloud application and its integration hooks. As a result, institutions not already using Nextcloud would need to deploy and maintain it to adopt NextPyter.

Deployment requires an initial configuration phase (e.g., reverse proxy, authentication integration, and policy configuration). Consequently, at least one technically skilled administrator is needed to bootstrap and maintain the system, particularly in institutional settings where security constraints, identity federation, and governance over execution environments are required.

In addition to addressing these adoption barriers, ongoing work targets broader execution back-end support, including integration with HPC- and research-oriented infrastructures such as Slurm and Apptainer, with the goal of extending NextPyter beyond conventional container runtimes while preserving the same user-facing workflow.

Finally, it should be noted that the evaluation results discussed in this work are based on two limited cohorts of students (16 first-year and 13 s-year students); future work will extend the study to larger and more heterogeneous user groups in order to strengthen the generalizability of the findings.

6. Conclusions and Future Work

This paper presented NextPyter, an open-source platform integrating Jupyter Notebooks with Nextcloud to support real-time collaborative learning in institutional educational settings. The work reported here documents both the platform's evolution from a research-oriented tool into a classroom-ready environment and its systematic evaluation through real teaching activities, representing the primary contributions of this study.

The evaluation, conducted across two university-level programming courses involving first-year and second-year Computer Engineering students, produced converging quantitative and qualitative evidence of the platform's relevance and usefulness. The analysis, focused specifically on the PU construct of the TAM, indicates a unanimously positive perception of NextPyter's usefulness, with composite PU scores of 4.09 and 3.53 on a 5-point scale and high internal consistency across both cohorts. Qualitative findings reinforce these results, revealing that students primarily envision the platform as a shared workspace supporting peer interaction, collaborative coding, project-based activities, and experiential learning.

From an educational perspective, the classroom evaluation provides preliminary insights into the activities for which students perceive an integrated collaborative programming environment as useful, particularly group-based exercises, shared problem-solving, and practical programming tasks. These findings describe students' perceptions of the platform's educational value rather than demonstrating improvements in learning outcomes.

Beyond its potential role in teaching and collaborative learning, NextPyter addresses a set of organizational and technical requirements that are particularly relevant in institutional educational environments. Its open-source and on-premises-oriented architecture can support greater control over data governance, privacy, and infrastructure management, while reducing reliance on proprietary external services. The integration of Jupyter Notebooks with Nextcloud also brings computational activities and document management into a single environment, reducing fragmentation between tools that are typically used separately in programming courses. Importantly, these characteristics were not evaluated only as architectural or conceptual properties: the platform was deployed in authentic university classroom activities, providing practical evidence of its feasibility in real teaching settings. The contribution of NextPyter can therefore be framed not primarily as a tool proven to improve learning, but as an institutional platform that combines collaborative computational work, integrated file management, and greater control over data and infrastructure, while showing positive perceived usefulness among students.

The study also identified areas for improvement. Students suggested the addition of structured communication features, such as dedicated comment sections, pointing to the importance of integrating discussion and feedback mechanisms alongside shared editing. From a technical standpoint, the current dependency on Nextcloud and the need for an initial administrative configuration phase represent adoption barriers that future development will seek to reduce.

Future Developments

Future developments will focus on four complementary directions, prioritizing the empirical and pedagogical gaps identified in this study. First, we plan to conduct extended evaluations involving larger and more diverse cohorts across multiple disciplines and institutions, moving beyond the exploratory scope of the present work. These studies will incorporate comparison conditions (e.g., standard Jupyter, Google Colab, or conventional individual instruction) to enable objective assessment of NextPyter's distinctive integration of collaborative editing, document management, and institutional data governance. They will include direct measures of learning outcomes, such as pre/post

programming assessments and code-quality metrics, to move from perceived utility to evidenced educational effectiveness. Second, we intend to involve educators directly in the evaluation process, collecting structured feedback on pedagogical integration, classroom management, and adoption barriers alongside student data, thereby capturing the dual perspective necessary for sustainable institutional uptake. Third, we are extending the daemon to support additional execution backends, including HPC-oriented infrastructures such as Slurm and Apptainer, to facilitate adoption in environments where resource allocation and security policies differ from single-host container deployments. Finally, we plan to improve administrative governance mechanisms (image management, resource quotas, session lifecycle rules) and to develop a monitoring layer for runtime characterization and energy-consumption tracking, supporting sustainable, evidence-based platform optimization.

Author Contributions: Conceptualization, L.B., F.F. and M.C.; methodology, L.B., F.F., M.C. and C.C.; software, F.F.; validation, L.B., F.F., M.C. and C.C.; formal analysis, L.B., F.F., M.C. and C.C.; investigation, M.J.A. and C.M.; resources, L.B., F.F., C.M., M.C. and C.C.; data curation, L.B., F.F. and M.C.; writing—original draft preparation, L.B., F.F., C.M., M.J.A. and C.C.; writing—review and editing, L.B., F.F., C.M., M.J.A., M.C. and C.C.; visualization, L.B., F.F., C.M. and M.C.; supervision, C.C.; project administration, L.B., M.C. and C.C.; funding acquisition, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of Modena and Reggio Emilia—Fondazione di Modena Project "NextPyter: Open-Source Collaborative Platform for Interdisciplinary Research" funded by Fondo di Ateneo per la ricerca Anno 2024—Bando per il finanziamento di progetti di ricerca interdisciplinari.

Institutional Review Board Statement: According to the EU General Data Protection Regulation (GDPR, Regulation (EU) 2016/679, Recital 26), anonymized data do not constitute personal data and therefore fall outside the scope of data protection regulations. The same principle is reflected in the Italian Privacy Code (Legislative Decree 196/2003, as amended by Legislative Decree 101/2018). Additionally, the survey platform used was self-hosted by our institution, meaning that no third-party services were involved in the collection or processing of the data. This further ensured that the data remained fully within the institution's control, enhancing privacy and security for participants. Moreover, in line with GDPR guidelines, universities and educational institutions have some flexibility in conducting surveys and research, especially if the data is fully anonymized. This grants us certain privileges in terms of processing and conducting surveys more freely, as the data are entirely under our control and fully anonymized. Finally, in line with Italian regulations and standard research practice, studies based exclusively on fully anonymized survey data do not require ethical approval by an institutional review board or ethics committee.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The raw data supporting the conclusions of this article cannot be shared due to ethical and privacy issues.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Questionnaire

NextPyter evaluation

1. What is your gender?

◯ Male ◯ Female ◯ Prefer not to specify

2. What is your role?

◯ Bachelor's student in Computer Science
◯ Bachelor's student in Computer Engineering
◯ Master's student in Computer Science
◯ Master's student in Computer Engineering
◯ Other

3. Have you ever felt the need to use collaborative tools in the classroom?

◯ Yes ◯ No

Perceived Usefulness (PU)

4. Do you think NextPyter could be useful for learning in your school/university?

◯ Yes ◯ No

5. Using NextPyter for learning would allow me to complete tasks more quickly.

Unlikely ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 Likely

6. Using NextPyter would improve my academic performance.

Unlikely ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 Likely

7. Using NextPyter would increase my productivity in completing school/university tasks.

Unlikely ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 Likely

8. Using NextPyter would enhance my learning effectiveness.

Unlikely ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 Likely

9. Using NextPyter would make it easier to carry out study activities and complete school/university assignments.

Unlikely ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 Likely

10. I would find NextPyter useful for my studies.

Unlikely ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 Likely

11. If yes, how do you think you would use NextPyter?

_____

Open feedback

This section is dedicated to open feedback, where you can share observations, suggestions, or personal comments in response to the questions provided.
These qualitative contributions will help complement the quantitative analysis with deeper and more contextual insights.

12. How could we improve the experience with NextPyter?

_____

13. Do you have any additional observations or comments regarding the use of the tool?

_____

Thank you for completing the questionnaire!

References

Attard, C., Berger, N., & Mackenzie, E. (2021). *The positive influence of inquiry-based learning teacher professional learning and industry partnerships on student engagement with stem* (Vol. 6). Frontiers Media SA. Available online: https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2021.693221 (accessed on 12 January 2026).

Bisong, E. (2019). *Building machine learning and deep learning models on google cloud platform*. Springer.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. [CrossRef]

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. Weerdmeester (Eds.), *Usability evaluation in industry* (1st ed., pp. 189–194). CRC Press. [CrossRef]

Campbell, E. C., Christensen, K. M., Nuwer, M., Ahuja, A., Boram, O., Liu, J., Miller, R., Osuna, I., & Riser, S. C. (2024). Cracking the code: An evidence-based approach to teaching Python in an undergraduate earth science setting. *Journal of Geoscience Education*, *73*, 239–258. Available online: https://www.tandfonline.com/doi/full/10.1080/10899995.2024.2384338 (accessed on 12 January 2026). [CrossRef]

Cristiano, K., & Triana, D. (2019). Google classroom as a tool-mediated for learning. In *Journal of physics: Conference series* (Vol. 1161, p. 012020). IOP Publishing.

Davis, F. D. (1989). Technology acceptance model: TAM. *Al-Suqri, MN, Al-Aufi, AS: Information Seeking Behavior and Technology Adoption*, *205*(219), 5.

Deepnote. (2026). *Deepnote for education*. Available online: https://deepnote.com/docs/edu-overview (accessed on 23 April 2025).

Deloitte. (2025). *Stem observatory 2024: Empowering the multiple transitions through stem skills* (3rd ed.). Available online: https://www.deloitte.com/content/dam/assets-zone2/it/it/docs/issues/work/2024/report-terza-edizione-osservatorio-stem.pdf (accessed on 12 January 2026).

Dillenbourg, P. (1999). *What do you mean by collaborative learning*. Available online: https://api.semanticscholar.org/CorpusID:54180662 (accessed on 15 January 2026).

Docker. (2026). *Docker: Accelerated container application development*. Available online: https://www.docker.com/ (accessed on 23 April 2025).

Dooley, R., Brandt, S. R., & Fonner, J. (2018). The Agave Platform: An open, science-as-a-service platform for digital science. In *Proceedings of the practice and experience on advanced research computing: Seamless creativity* (pp. 1–8). ACM Digital Library.

Du, D., Baird, T., Bonella, S., & Pizzi, G. (2022). OSSCAR, an open platform for collaborative development of computational tools for education in science. *arXiv*, arXiv:2203.12019. [CrossRef]

European Commission and Directorate-General for Education, Youth, Sport and Culture. (2025). *Education and training monitor 2025—Comparative report*. Publications Office of the European Union. [CrossRef]

European Commission and Directorate-General for Research and Innovation. (2015). *Science education for responsible citizenship–Report to the european commission of the expert group on science education*. Publications Office. [CrossRef] [PubMed]

Faenza, F., Canali, C., & Carbonaro, A. (2021a). ICT extra-curricular activities: The "Digital Girls" case study for the development of human capital. In A. Visvizi, O. Troisi, & K. Saeedi (Eds.), *Research and innovation forum 2021*. Springer International Publishing.

Faenza, F., Canali, C., Colajanni, M., & Carbonaro, A. (2021b). The digital girls response to pandemic: Impacts of in presence and online extracurricular activities on girls future academic choices. *Education Sciences*, *11*(11), 715. [CrossRef]

Faenza, F., Canali, C., Fregni, L., & Maccaferri, E. (2024). NextPyter: Open-source research collaborative platform. In *Practice and experience in advanced research computing 2024: Human powered computing* (pp. 1–9). Association for Computing Machinery.

Faenza, F., Casari, M., Burchiellaro, L., & Canali, C. (2026). *Nextpyter: Collaborative notebook environment for education and research*. Zenodo. [CrossRef]

Felber, N., Lipworth, W., Tian, A., Schwab, D., & Wangmo, T. (2024). Informing existing technology acceptance models: A qualitative study with older persons and caregivers. *European Journal of Ageing*, *21*(1), 12. [CrossRef] [PubMed]

Fernández, L., Andersson, R., Hagenrud, H., Korhonen, T., Laface, E., & Zupanc, B. (2016, May 8–13). *Jupyterhub at the ess. an interactive python computing environment for scientists and engineers*. IPAC2016, Busan, Republic of Korea.

Fishbein, M., & Ajzen, I. (1975). *Belief, attitude, intention and behaviour: An introduction to theory and research* (Vol. 27). Addison-Wesley.

Grandell, L., Peltomaki, M., Back, R.-J., & Salakoski, T. (2006). Why complicate things?: Introducing programming in high school using Python. In *Acm international conference proceeding series* (Vol. 165, pp. 71–80). ACM International.

Ješková, Z., Lukáč, S., Šnajder, L., Guniš, J., Klein, D., & Kireš, M. (2022). Active learning in STEM education with regard to the development of inquiry skills. *Education Sciences*, *12*(10), 686. [CrossRef]

Jupyter Notebook. (2026). Available online: https://jupyter.org/ (accessed on 23 April 2026).

Kaggle. (2026). *Kaggle: Your machine learning and data science community*. Available online: https://www.kaggle.com/ (accessed on 23 April 2025).

Kelter, R., Kramer, M., & Brinda, T. (2019). Teachers' perspectives on learning and programming environments for secondary education. In *Empowering learners for life in the digital age: IFIP TC 3 open conference on computers in education, OCCE 2018, Linz, Austria, 24–28 June 2018, revised selected papers* (pp. 47–55). Springer.

Khosravi, H., Kitto, K., & Williams, J. J. (2019). RiPPLE: A Crowdsourced adaptive platform for recommendation of learning activities. *Journal of Learning Analytics*, *6*(3), 91–105. Available online: https://learning-analytics.info/index.php/JLA/article/view/6373 (accessed on 13 January 2026). [CrossRef]

King, W. R., & He, J. (2006). A meta-analysis of the technology acceptance model. *Information & Management*, *43*(6), 740–755. Available online: https://www.sciencedirect.com/science/article/pii/S0378720606000528 (accessed on 13 January 2026). [CrossRef]

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., & Kelley, K. (2016). Jupyter notebooks–a publishing format for reproducible computational workflows. In *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS press.

Legarde, M. A. A., Mundo, M. D., Chavez, J. V., Jailani, A. B., Calzada, K. P. D., Quisay, A. R. C., & Piñero-Abdurajak, K. P. (2025). Empowering learning engagement in higher education with active learning experiences in STEM classrooms. *Environment and Social Psychology*, *10*(9), ESP-3878. [CrossRef]

Lund, A. (2001). Measuring usability with the USE questionnaire. *Usability and User Experience Newsletter of the STC Usability SIG*, *8*(2), 3–6.

Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. In *Proceedings of the 7th baltic sea conference on computing education research* (pp. 112–123). ACM.

Nardito, L. M., Llanto, M. H., Ebuenga, J. C., Deladia, J. D., Manilag, M. E. F., Rejuso, R. M., & Digo, G. S. (2025). Assessing the efficacy of Google Classroom as an instructional medium in blended learning environments: A systematic review and meta-synthesis. *International Journal of Multidisciplinary Research and Growth Evaluation*, *6*(4), 582–593. [CrossRef]

NextCloud. (2026). *Nextcloud—Open source content collaboration platform*. Available online: https://nextcloud.com/ (accessed on 23 April 2026).

Observable. (2026). *Observable: The javascript notebook for data visualization*. Available online: https://observablehq.com/ (accessed on 23 April 2025).

OECD Publishing, Paris. (2023). *PISA 2022 results (volume I): The state of learning and equity in education*. OECD Publishing.

Pedaste, M., Mäeots, M., Siiman, L. A., de Jong, T., van Riesen, S. A. N., Kamp, E. T., Manoli, C., Zacharia, Z. C., & Tsourlidaki, E. (2015). Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational Research Review*, *14*, 47–61. Available online: https://api.semanticscholar.org/CorpusID:35300720 (accessed on 12 January 2026). [CrossRef]

Project Jupyter. (2026). *Binder: Reproducible, shareable, interactive computing*. Available online: https://jupyter.org/binder (accessed on 23 April 2025).

Replit. (2026). *Replit: The ai-powered software development platform*. Available online: https://replit.com/ (accessed on 23 April 2026).

Rocard, M., Csermely, P., Jorde, D., Lenzen, D., Walberg-Henriksson, H., & Hemmo, V. (2007). *Science education now: A renewed pedagogy for the future of Europe. European Commission*. Publications Office.

Sayeed, S., Saiem, B., Sany, A.-A., Sharmin, S., & Islam, A. B. M. A. A. (2025). CollaClassroom: An AI-augmented collaborative learning platform with LLM support in the context of Bangladeshi university students. *arXiv*, arXiv:2511.11823. [CrossRef]

Sælemyr, K., & Bjørndal, J. E. (2019). «Utflukter sitter lengre i hjernen». Elevers synspunkter på hvordan de lærer naturfag: «Excursions remains longer in the brain». Student's views on teaching methods in science education. *Nordic Studies in Science Education*, *15*(3), 226–241.

Scherer, R., & Teo, T. (2019). Unpacking teachers' intentions to integrate technology: A meta-analysis. *Educational Research Review*, *27*, 90–109. Available online: https://www.sciencedirect.com/science/article/pii/S1747938X18303622 (accessed on 15 January 2026). [CrossRef]

Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017). Construction of a benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, *4*(4), 40–44. [CrossRef]

Stahl, G., Koschmann, T., & Suthers, D. D. (2006). *Computer-supported collaborative learning: An historical perspective*. Available online: https://api.semanticscholar.org/CorpusID:8868608 (accessed on 13 January 2026).

Tufino, E., Oss, S., & Alemani, M. (2024). Integrating Python data analysis in an existing introductory laboratory course. *European Journal of Physics*, *45*(4), 045707. Available online: https://iopscience.iop.org/article/10.1088/1361-6404/ad4fcc (accessed on 15 January 2026). [CrossRef]

Vallejo, W., Díaz-Uribe, C., & Fajardo, C. (2022). Google colab and virtual simulations: Practical e-learning tools to support the teaching of thermodynamics and to introduce coding to students. *ACS Omega*, *7*(8), 7421–7429. [CrossRef] [PubMed]

Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, *39*(2), 273–312. [CrossRef]

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, *46*(2), 186–204. [CrossRef]

Venkatesh, V., Morris, M., Davis, G., & Davis, F. (2003). User Acceptance of information technology: Toward a unified view. *MIS Quarterly*, *27*, 425–478. [CrossRef]

Vygotsky, L. S. (1978). *Mind in society: Development of higher psychological processes*. Harvard University Press. Available online: http://www.jstor.org/stable/j.ctvjf9vz4 (accessed on 10 April 2026).

Weigend, M. (2020). Programming languages for secondary schools, Python. In *Encyclopedia of education and information technologies* (pp. 1370–1378). Springer.

Withers, A., Bockelman, B., Weitzel, D., Brown, D., Gaynor, J., Basney, J., Tannenbaum, T., & Miller, Z. (2018). SciTokens: Capability-based secure access to remote scientific data. In *Proceedings of the practice and experience on advanced research computing: Seamless creativity* (pp. 1–8). ACM Digital Library.

Xinh, L. T., Trung, D. N., Hong, B. V., Luong, N. T., & Nga, L. T. M. (2025). Organizing collaborative learning in STEM education to develop students' 21st-century skills. In *2025 10th international stem education conference (iSTEM-Ed)* (pp. 1–6). IEEE. [CrossRef]

Zhao, Z., Koulouzis, S., Bianchi, R., Farshidi, S., Shi, Z., Xin, R., Wang, Y., Li, N., Shi, Y., Timmermans, J., & Kissling, W.D. (2022). Notebook-as-a-VRE (NaaVRE): From private notebooks to a collaborative cloud virtual research environment. *Software: Practice and Experience*, *52*(9), 1947–1966. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.