

From Pods to Prompts: A Pilot Evaluation of NRP-Based Kubernetes, Open WebUI, and LLM Fine-Tuning Workflows

Sriram Srinivasan
sriram882009@gmail.com

Hamdan Alabsi
halabsi@bowiestate.edu

Rand Obeidat
robeidat@bowiestate.edu

Ali Raza
araza@bowiestate.edu

Qingqing Sun
qsun@bowiestate.edu

Kavita Kapur
KKAPUR@bowiestate.edu

Bowie State University
College of Business
Bowie, MD 20715, USA

Abstract

This pilot study evaluates a scaffolded instructional framework that uses the National Research Platform (NRP) to teach students how to deploy, configure, fine-tune, and evaluate open-source Large Language Models within cloud-native Kubernetes environments. The framework consists of nine hands-on modules covering NRP onboarding, Kubernetes pod, resource management, networking, persistent storage, vLLM hosting, Open WebUI interaction, fine-tuning workflows, and responsible AI assessment. Implemented with a cohort of 45 Science, Technology, Engineering, and Mathematics (STEM) students, the modules enabled learners to work with authentic infrastructure components, including namespaces, pods, services, ingress rules, persistent volume claims, GPU accelerators, and OpenAI-compatible API endpoints. Results show high completion across the scaffolded workflow, strong student satisfaction, and measurable competency development across infrastructure and model-evaluation domains. The greatest implementation challenges occurred in Kubernetes resource management, networking, fine-tuning, and full vLLM integration, indicating where additional scaffolding is needed. The study demonstrates that NRP-supported infrastructure can lower cost and access barriers for hands-on generative AI education while helping students move beyond black-box prompt use toward practical AI infrastructure engineering.

Keywords: Cloud-Native AI Education, Scaffolded teaching, AI Pedagogy, Assessment frameworks

From Pods to Prompts: A Pilot Evaluation of NRP-Based Kubernetes, Open WebUI, and LLM Fine-Tuning Workflows

Sriram Srinivasan, Hamdan Alabsi, Rand Obeidat, Ali Raza, Qingqing Sun and Kavita Kapur

1. INTRODUCTION

Large Language Models (LLMs) (Minaee et al., 2024) have rapidly become ubiquitous across higher education, industry, and daily life. While most users interact with these models exclusively through conversational chat interfaces, this standard paradigm hides the underlying cloud-native infrastructure encompassing Kubernetes, containerized workloads, networking, persistent storage, and API endpoints behind a black box. For Information Systems (IS) students and educators, developing a deep understanding of both LLM behavior and this supporting infrastructure is imperative, especially as organizations increasingly deploy domain-specific models to maximize task efficiency.

Recognizing this educational shift, major accreditation agencies emphasize that computing and business education must evolve. ABET (2026) mandates under its Computing Accreditation Commission criteria that students demonstrate a measurable ability to analyze complex computing problems and to design, implement, and evaluate structured, computing-based solutions. Meeting ABET standards requires that future professionals move past superficial application use; they must master the structural cloud-native infrastructure, resource provisioning parameters, and configuration steps that dictate how modern computing environments serve and adapt advanced AI models.

However, a primary bottleneck in higher education prevents widespread hands-on mastery: the limited availability and prohibitive cost of the graphics processing units (GPUs) required to run these workloads. To bridge this accessibility gap, this paper demonstrates how higher education institutions can leverage the NRP to access high-performance computing resources and accelerators at zero cost. Funded by the National Science Foundation, the NRP operates as a distributed, multi-tenant scientific Kubernetes cluster spanning dozens of institutions nationwide (Weitzel et al., 2025). This shared infrastructure connects researchers and educators to advanced high-performance computing resources, large-scale data storage pools, and specialized GPU accelerators, intentionally serving as an open testbed to transform technical classroom instruction (Weitzel et al., 2025). We introduce a portable, hands-on pedagogical framework consisting of specialized AI modules that enable students to deploy, configure, fine-tune, and customize open-source LLMs. These modules guide students through authentic engineering tasks: authenticating with the NRP via federated university credentials, joining a course-assigned Kubernetes namespace, and requesting dedicated accelerator resources.

Once connected to the NRP, students explore model deployment through two distinct modalities. First, they interact with an NRP-hosted Open WebUI instance, which provides an accessible interface similar to commercial platforms like ChatGPT for baseline model evaluation. Second, they utilize Application Programming Interfaces (APIs) to programmatically connect to their hosted models, adjust inference parameters, and build custom applications suited to specific use cases.

We implemented these modules within a cohort of 45 STEM students who possessed a foundational background in Python programming. The laboratory exercises were highly successful, majority of the cohort was able to use the NRP resources to complete the course assessments. Ultimately, this study establishes the feasibility of utilizing the NRP for scalable classroom instruction, bypassing the restrictive resource constraints and cost caps typically associated with commercial cloud platforms like AWS and GCP. By demystifying the real-world mechanics of Kubernetes, pods, storage, and networking, these modules offer a flexible, ready-to-adopt curriculum for IS and related computing disciplines.

1.1 Research Questions(RQs)

Based on the current landscape of AI and the growing demand for graduates who have a clear understanding of GenAI systems, how LLMs work, and the cloud-native infrastructure to run LLMs, this study addresses four primary research questions:

RQ1. To what extent can a cohort successfully complete a scaffolded NRP-based workflow involving Kubernetes, pod management, networking, Open WebUI, and LLM experimentation?

RQ2. Which workflow components present the greatest implementation challenges during execution of the NRP-based workflow?

RQ3. What competency levels are demonstrated across the NRP workflow domains, including Kubernetes, pod management, networking, Open WebUI, LLM adaptation, and responsible AI assessment?

RQ4. How effectively can a cohort integrate cloud-native infrastructure concepts and LLM workflows within the final capstone experiment?

1.2 Contributions

This work delivers three primary contributions to the fields of IS education and cloud-native curriculum design, as outlined in the research framework in Figure 1 alongside the core research questions. First, it establishes an adaptable, zero-cost pedagogical framework that enables resource-constrained institutions lacking local hardware accelerators to teach LLM mechanics and cloud-native AI architecture across diverse computing and data analytics courses. Second, this study presents a rigorous empirical evaluation of the framework with a cohort of 45 STEM students, delivering concrete evidence of scalability by analyzing workflow completion frequencies, measured competency gains, and specific infrastructure hurdles while documenting a substantial increase in student self-efficacy regarding LLM deployment via post assessment reflections. Third, the authors open-source the complete set of nine specialized, hands-on laboratory modules, including all foundational source code, configuration templates, and exercise handouts, to facilitate immediate adoption and integration by other educators.

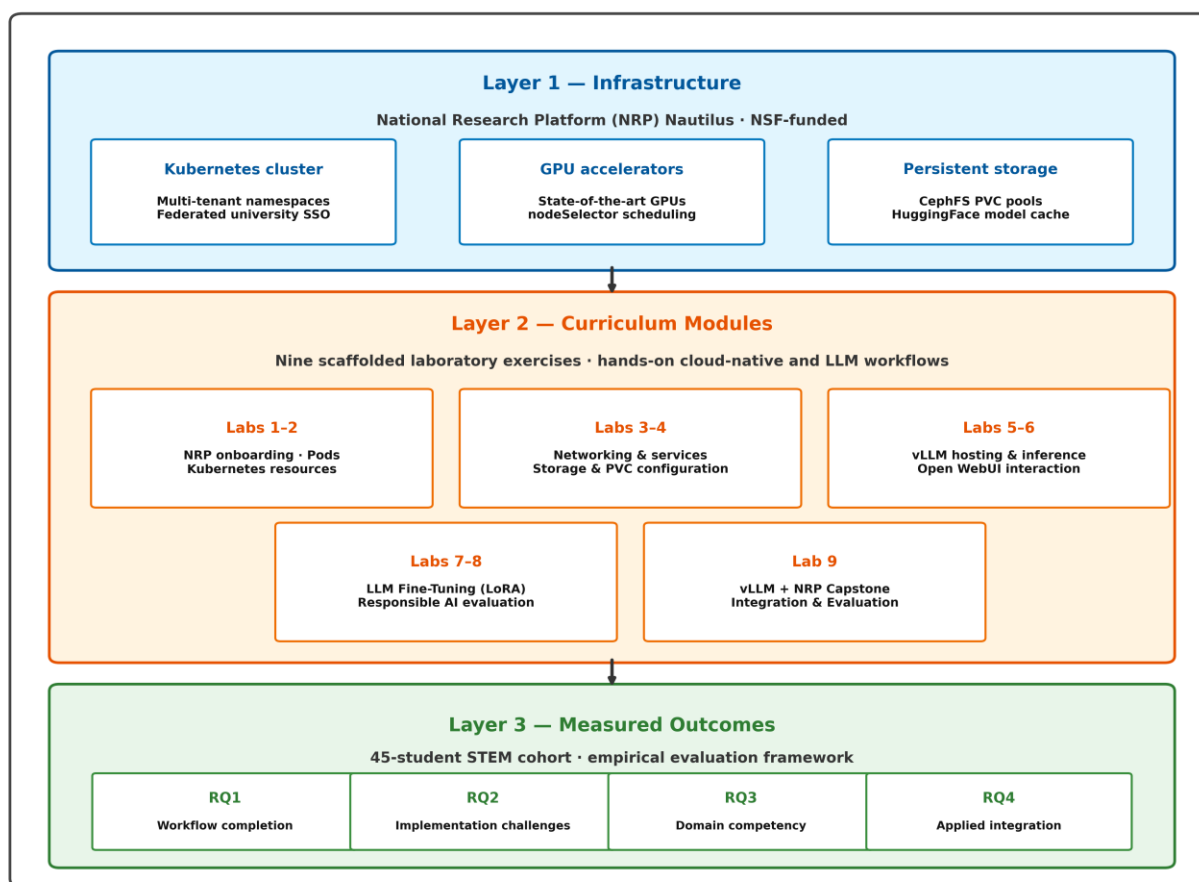


Figure 1. NRP-based framework linking infrastructure, scaffolded AI labs, and measured workflow outcomes.

2. BACKGROUND & RELATED WORK

This section establishes the foundational context for the study by reviewing cloud-native AI infrastructure, examining the current landscape of cloud and AI education, and identifying the specific research gap this study addresses within IS curricula. Finally, it details the theoretical frameworks that ground our instructional and assessment design.

2.1 Cloud-Native Infrastructure & AI Education

Cloud-native infrastructure provides the operational foundation for modern AI systems by orchestrating containers, microservices, and declarative APIs. Kubernetes automates the deployment and scaling of these workloads, but managing GPU accelerators remains a premier operational challenge due to fragmentation risks. Traditional cloud computing curricula emphasize using cloud technologies as turnkey solutions, whereas industry demand requires graduates who can actively engineer and manage these architectures. While existing literature extensively analyzes the consumption of AI, it largely ignores the technical infrastructure required to host, deploy, and configure these models. To shift students from passive consumers to active developers, educational frameworks must provide transparent access to underlying hardware and orchestration layers. (An extended review of the cloud-native AI infrastructure literature is provided in Appendix A).

2.2 Theoretical Grounding

This study is grounded in Bandura's Self-Efficacy Theory and Kolb's Experiential Learning Model. The curriculum develops self-efficacy through progressive mastery experiences, transitioning from guided labs to independent capstone execution, which we operationalized via continuous student reflections. Kolb's model structures the precise hands-on mechanics, moving from concrete experience (interfacing with the NRP) to active experimentation (end-to-end integration).

2.3 Research Gap

Collectively, our preliminary literature survey exposes a critical pedagogical gap: while existing literature extensively analyzes the consumption and ethical challenges of AI, it largely ignores the technical infrastructure required to host, deploy, and configure these models, leaving limited empirical work evaluating integrated, Kubernetes-supported LLM workflows within IS environments. To move students from passive consumption to active engineering, educational frameworks must provide transparent access to the underlying hardware and orchestration layers. This study directly addresses this gap by shifting the educational focus from prompt engineering to AI infrastructure engineering. By leveraging the NRP, we show how educators can bypass traditional commercial cloud constraints, giving students direct, hands-on control over Kubernetes namespaces, storage, pods, and dedicated accelerators to deploy and customize open-source LLMs.

2.4 Theoretical Grounding

Bandura's Self-Efficacy Theory (Bandura, 1977) establishes that an individual's confidence in their ability to perform a task directly drives their motivation, persistence, and ultimate performance. The curriculum workflow structurally develops this self-efficacy through three progressive phases: students first complete guided labs where step-by-step instructions clarify both the execution and the underlying reasoning for each technical action; they then transition to making controlled modifications in the environment to validate their understanding as they master core concepts; and they finally complete an integrated lab where they independently apply all cumulative module concepts to execute a final, minimal assisted experiment.

Kolb's Experiential Learning Model (Kolb, 1984) provides a secondary lens by ensuring students learn through direct, hands-on engagement rather than passive observation, mapping cleanly across the four stages of the experiential learning cycle. Students first gain concrete experience by actively interfacing with the NRP to request resources, deploy pods, configure networking, interact with Open WebUI, and execute LLM workflows. They move to reflective observation by using APIs and logs to monitor real-time system performance metrics including memory bandwidth, latency, GPU power, and memory utilization before practicing abstract conceptualization by connecting these operational observations to

broader theoretical topics like orchestration, resource allocation, service exposure, model serving, fine-tuning, hallucination, and responsible AI. They complete the cycle through active experimentation in the final module, synthesizing these concepts to solve novel technical problems. Combined, these two models form a cohesive theoretical foundation for the study's assessment design, where Bandura's framework measures shifts in student confidence and perception across scaffolded tasks, while Kolb's model structures the precise hands-on mechanics and final integration activities required for true experiential learning. For this initial pilot, self-efficacy is operationalized through continuous student reflections posted to Discord after each module assessment, capturing real-time shifts in their technical confidence and specific newly acquired learnings as they executed the scaffolded workflow.

3. METHODOLOGY

This section describes the design, implementation, and evaluation of the NRP-based cloud-native AI workflow. First, it presents the NRP workflow architecture. Next, it explains the assessment framework used to evaluate workflow completion, academic performance, domain-specific competency, and final integration. The section then describes the data sources collected for this study. Finally, it outlines the analysis strategy used to answer the study's four research questions.

3.1 NRP Workflow Architecture

The study utilized the NRP Nautilus cluster as the computational environment for the cloud-native AI workflow, assigning the cohort to a dedicated Kubernetes namespace to execute their deployments. Figure 2 presents a high-level overview of this architecture, illustrating the connections between the backend compute resources and the user-facing interfaces. The comprehensive technical exposition of each infrastructural layer depicted in Figure 2, including Kubernetes resource management, pod orchestration, service routing, persistent storage, vLLM model serving, and Open WebUI interaction is fully detailed in Appendix B.

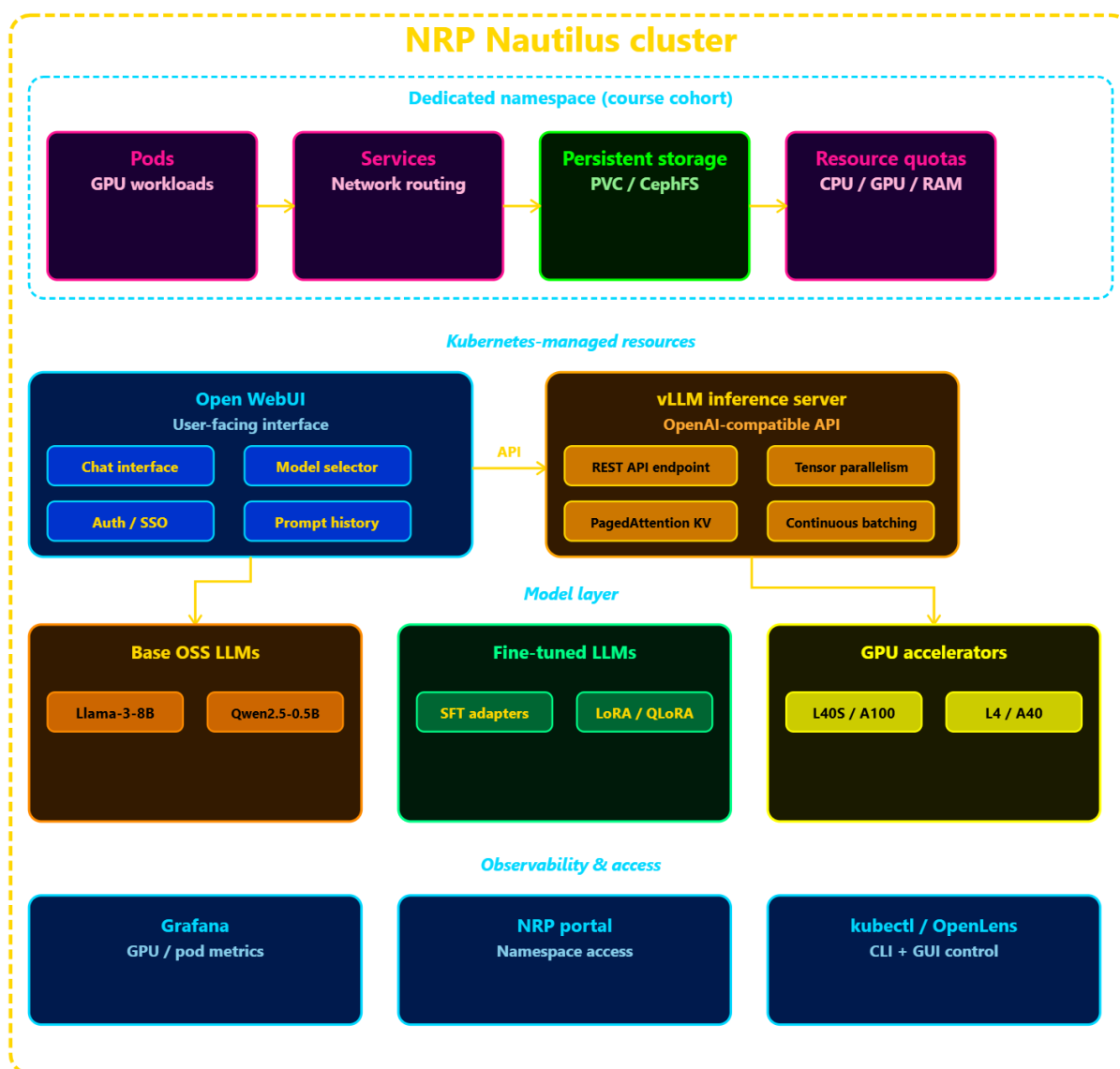


Figure 2. NRP Workflow Architecture.

3.2 Assessment Framework

This study evaluates how the cohort managed workflow completion, academic performance across all nine modules, domain-specific competency, and the structural integration of multiple core concepts during the final comprehensive lab. The assessment framework was formed based on Bandura's self-efficacy theory and Kolb's experiential learning model. Bandura's framework supports the use of completion, performance, and integration outcomes as evidence of mastery experiences across increasingly complex technical tasks, which we operationalize by measuring the cohort's retrospective, self-reported confidence in mastering these domains. Kolb's framework supports the design of the workflow as an experiential sequence in which the cohort completed concrete technical activities, observed system behavior, connected those observations to infrastructure and AI concepts, and applied the concepts through active experimentation.

Table C1 in Appendix C outlines the structured assessment framework used in the study. The framework identifies the operational criteria, evidence sources, and theoretical alignment used to evaluate each dimension.

3.3 Assessment Data Sources

To ensure analytical validity, we gathered empirical data from seven distinct sources throughout the pilot. All data collection procedures, including the use of student academic records and course evaluations, were reviewed and granted exemption by the Institutional Review Board at Bowie State University. These sources include automated lab completion logs that tracked student progress through the onboarding and deployment pipeline, and formal assignment grades that measured individual technical accuracy. We also utilized standardized rubric scores to assess conceptual mastery, final project evaluation scores that captured end-to-end system orchestration, and direct artifact evaluations of student-generated configuration files, YAML resource requests, and active endpoints. Finally, to operationalize Bandura's self-efficacy framework, we extracted qualitative reflection data from student posts on Discord following each module's assessment. These continuous reflections captured both the specific new concepts students learned and their self-reported confidence levels across the workflow domains. We also incorporated responses from the standard institutional end-of-course evaluation survey administered by Bowie State University, which we use in Section 5 to assess self-reported satisfaction and perceived learning gains.

3.4 Analysis Strategy

We analyzed quantitative completion rates (RQ1) and mapped assignment scores alongside qualitative reflection themes to calculate a Combined Workflow Difficulty Index (RQ2). We normalized rubric evaluations onto a standardized five-point scale to assess discrete domain competencies (RQ3) and holistic capstone integration (RQ4). The comprehensive analytical mapping, including formula calculations for the Difficulty Index and Competency Scores (Equation 1 in Appendix D).

4. WORKFLOW EVALUATION

This section presents the nine instructional modules. Each module included a hands-on laboratory activity designed to develop a specific component of the NRP-based cloud-native AI workflow. The nine-module workflow also aligns with the IS2020 competency model (Leidig & Salmela, 2021). The full module-to-competency alignment is provided in Table E1 in Appendix E, and Table F1 in Appendix F provides the links to access the modules. The core pedagogical objectives for each module are summarized below; the comprehensive, step-by-step technical execution details for all nine modules are preserved in Appendix G.

4.1 Module 1: NRP Onboarding and Infrastructure Fundamentals

This introductory module connects students to the core architecture of the NRP through account configuration, namespace registration, and initial resource provisioning (Figure 3), allowing them to master foundational cloud-native infrastructure concepts.

4.2 Module 2: Kubernetes Fundamentals and Containerized Resource Requests

This module introduces core principles of container orchestration, requiring students to provision and manage pods, jobs, and deployments by authoring precise declarative YAML files to request GPU, CPU, and memory allocations.

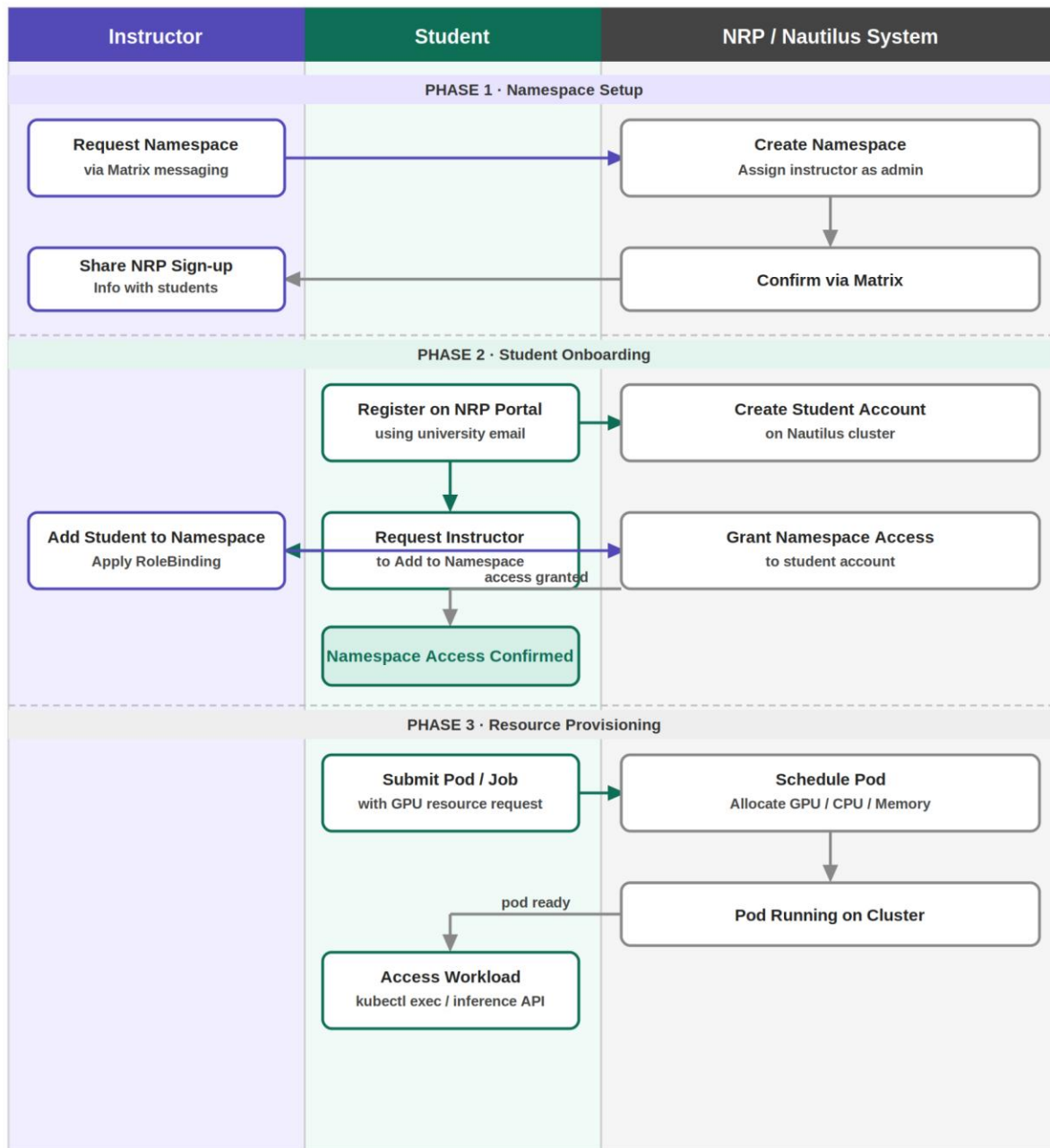


Figure 3. NRP Onboarding and Resource Provisioning Workflow.

4.3 Module 3: Networking and Microservice Communication

This module explores data communication within a Kubernetes cluster, focusing on how AI microservices interact by having students deploy an inference service and expose it via a Kubernetes service abstraction.

4.4 Module 4: Storage Infrastructure and Persistent Volumes

This module explores distributed storage options within the NRP, training students to configure and attach Persistent Volume Claims (PVCs) to active computing workloads to retain massive datasets and model weights across container lifecycles.

4.5 Module 5: LLM Hosting and Inference Engine Orchestration

This module covers the core mechanics of LLMs, demonstrating how to host, configure, and serve lightweight open-source architectures (e.g., Llama-3-8B or Qwen2.5-0.5B-Instruct) using the high-performance vLLM engine.

4.6 Module 6: Open WebUI Workflow and Prompt Engineering Evaluation

This module introduces Open WebUI, a user-facing interface, training students to connect to inference endpoints programmatically, construct structured prompt templates, and systematically evaluate output quality.

4.7 Module 7: LLM Fine-Tuning and Parameter-Efficient Adaptation

This module covers the principles of instruction fine-tuning, demonstrating how to adapt a base LLM using specialized instruction-response datasets and Parameter-Efficient Fine-Tuning (PEFT) techniques like LoRA.

4.8 Module 8: Responsible AI Evaluation and Algorithmic Guardrails

This module guides students through evaluating LLM outputs for critical vulnerabilities—including hallucinations, fairness, and systematic bias—culminating in a formal assessment report on algorithmic guardrails.

4.9 Module 9: vLLM Integration with NRP and Open WebUI

This final capstone experiment requires the cohort to synthesize all core curriculum concepts, integrating cloud-native infrastructure, vLLM model serving, Open WebUI configuration, and fine-tuned models into a single end-to-end pipeline. Because this comprehensive capstone tests all key concepts, Lab 9 serves as the primary assessment artifact for RQ4.

5. RESULTS

Each item was evaluated using a one-sample t-test comparing the observed mean against the scale midpoint (neutral response), with the null hypothesis that the true mean equals the midpoint. The quantitative feedback summarized in Table H1 in Appendix H relies on a standard institutional course evaluation instrument that was not designed specifically for this study. Consequently, response rates varied, yielding a Valid N ranging from 33 to 39 against the 45-student cohort. This variance indicates that between 13% and 27% of the cohort is missing from every row of the survey data. Despite this missing data, the responding students reported highly positive outcomes across the selected survey metrics, supporting the qualitative observations regarding the perceived value of the implemented AI modules. These responses suggest that the laboratory exercises were associated with perceived improvements in technical competency and overall learning experience. However, because these data are derived from a standard post-course evaluation, these findings represent evidence of student perception and course-level feasibility rather than causal evidence that the modules alone produced learning gains.

The data underscore the pedagogical value of integrating high-performance cloud infrastructure into the curriculum while also allowing for alternative interpretations, such as the influence of instructor support, peer collaboration, prior student interest, or the novelty of working with generative AI tools. When evaluating how laboratory experiences contributed to their understanding of course theories and concepts, the cohort provided a high mean score of 4.53 (SD = 0.79), with 88.2% of the 34 responding

students offering a positive evaluation ($p < .001$). Similarly, the 39 students who evaluated whether the course activities improved their computer technology skills yielded a mean score of 4.56 ($SD = 0.75$), with 89.7% reporting a positive outcome ($p < .001$). These self-reported outcomes align with students' reported increases in familiarity with fine-tuning models and navigating the NRP workflow. Directly reflecting Bandura's construct of self-efficacy, the continuous Discord reflections revealed distinct confidence gains; as students progressed through the modules, they reported greater confidence in critically evaluating model output limitations, including bias, hallucination, and fairness, and in explaining deployment concepts to their peers.

The student responses also provide support for the instructional design and the choice of technological frameworks. Participants rated the variety of technological tools used to enhance instruction highly, returning a mean score of 4.45 ($SD = 0.83$) and an 84.2% positive response rate from 38 respondents. Furthermore, the cohort expressed substantial satisfaction with how clearly the instructor explained the laboratory procedures ($M = 4.45$, $SD = 0.79$, 87.9% positive response). While some students noted initial challenges adjusting to the NRP infrastructure, managing YAML configurations, and monitoring containerized accelerator resources, these structural hurdles did not appear to diminish their educational engagement. Overall, this deployment was associated with strong satisfaction marks, with students giving a mean score of 3.51 ($SD = 0.64$) out of 4.00 for the course overall, representing a 92.3% positive response rate, and a mean score of 3.51 ($SD = 0.79$) out of 4.00 for the instructor's overall teaching performance.

5.1 Completion Outcomes (RQ1)

RQ1 examines the extent to which a cohort can successfully complete a scaffolded, NRP-based workflow. To evaluate this question, we tracked empirical completion frequencies across all nine scaffolded modules. The data reveal highly resilient completion metrics across the instructional pipeline, demonstrating that the scaffolded design successfully guided the cohort through complex cloud-native AI infrastructures.

As illustrated in Figure 4, the cohort achieved a perfect 100% completion rate during both Module 1 (NRP Setup) and Module 6 (Open WebUI Workflow). These baselines establish that the introductory onboarding and initial portal interactions posed minimal systemic barriers. Similarly, the cohort demonstrated strong proficiency in core resource hosting and management assignments, achieving a 98% completion rate in Module 5 (LLM Hosting and Access), a 96% completion rate in Module 8 (Responsible AI Evaluation), and a 93% completion rate in Module 4 (Storage and PVCs).

The data also reveal that the more advanced infrastructure orchestration and model adaptation tasks introduce heightened operational complexity, which subsequently resulted in minor performance drops. Specifically, Module 2 (Kubernetes and Pods) and Module 3 (Networking and Data Communication) yielded completion rates of 89% and 91%, respectively. The lowest completion thresholds occurred during the most technically demanding exercises, where Module 7 (LLM Fine-Tuning) dropped to 82% and Module 9 (vLLM, NRP, and Open WebUI integration) concluded at 84%. These trends confirm that advanced activities involving Kubernetes resource manipulation and direct model alignment require scaffolded support and additional instructional assistance to maximize completion success.

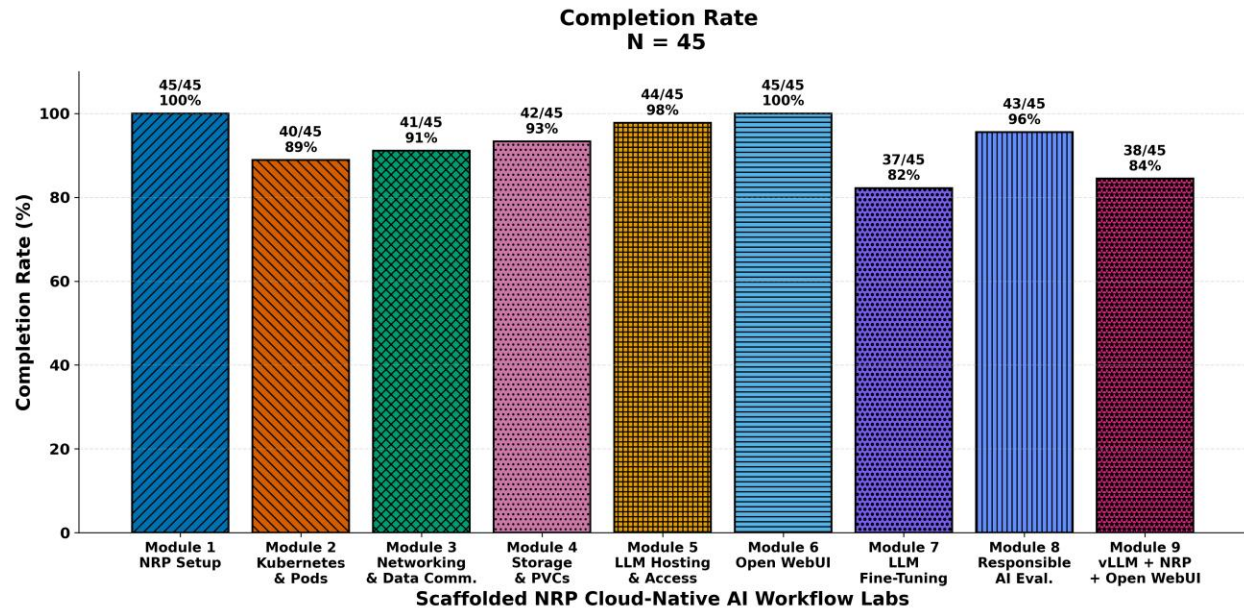


Figure 4. Completion Rates Across Scaffolded NRP Cloud-Native AI Workflow Modules.

5.2 Workflow Difficulty Analysis (RQ2)

RQ2 examined which workflow components exhibited the greatest implementation challenges and performance variation during execution. Section 3.4 details the evaluation methodology for this question.

Figure 5 illustrates the mean assignment scores alongside the associated performance variation across the cohort, where error bars represent one standard deviation from the mean. The quantitative results reveal stark operational disparities between infrastructure and application layers. NRP onboarding and Open WebUI configurations yielded the highest average scores and the lowest performance variance ($M = 94.71\%$, $SD = 3.34$), with most participants completing these tasks successfully with minimal assistance. Conversely, Kubernetes resource management, networking, LLM hosting, fine-tuning, and vLLM integration exhibited significantly lower average scores and substantially wider performance spreads ($M = 77.93\%$, $SD = 13.93$), frequently requiring intensive instructional intervention, Welch's $t(278.6) = 16.89$, $p < .001$, $d = 1.66$.

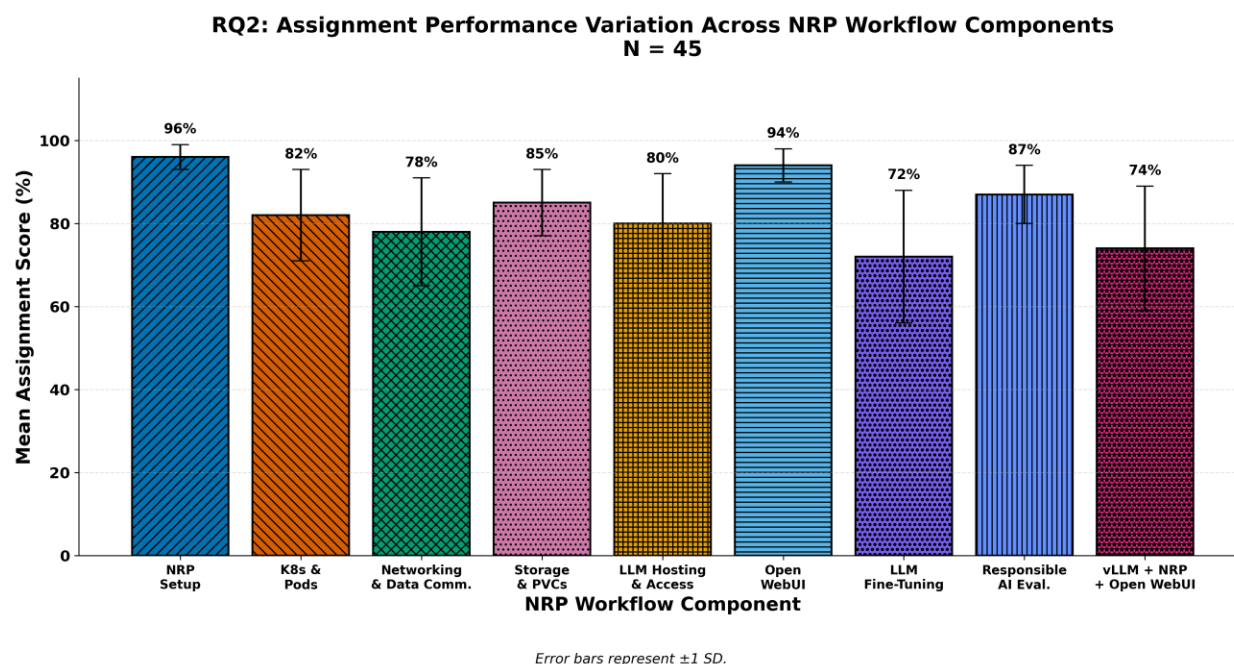


Figure 5. Assignment Performance Variation Across NRP Workflow Components.

Figure 6 summarizes the frequency of challenge-related themes extracted from participant reflections, reporting the total number of participants who explicitly cited difficulties per component. This qualitative friction data directly corroborates the empirical assignment outcomes. Fine-tuning, vLLM integration, Kubernetes resource management, and networking concepts generated the most challenge-related comments. Participants specifically reported difficulties with managing resource allocations, monitoring pod execution, configuring services, and interpreting model-adaptation parameters. Conversely, Open WebUI and NRP onboarding generated negligible student friction. Figure 7 presents a Combined Workflow Difficulty Index (Equation (2) discussed in Appendix D).

Ultimately, this analysis indicates that while participants rapidly adopt user-facing LLM application workflows, backend infrastructure-oriented activities introduce steep learning curves. These findings suggest that future iterations must allocate additional instructional time to Kubernetes core concepts, resource scheduling, networking, and model serving.

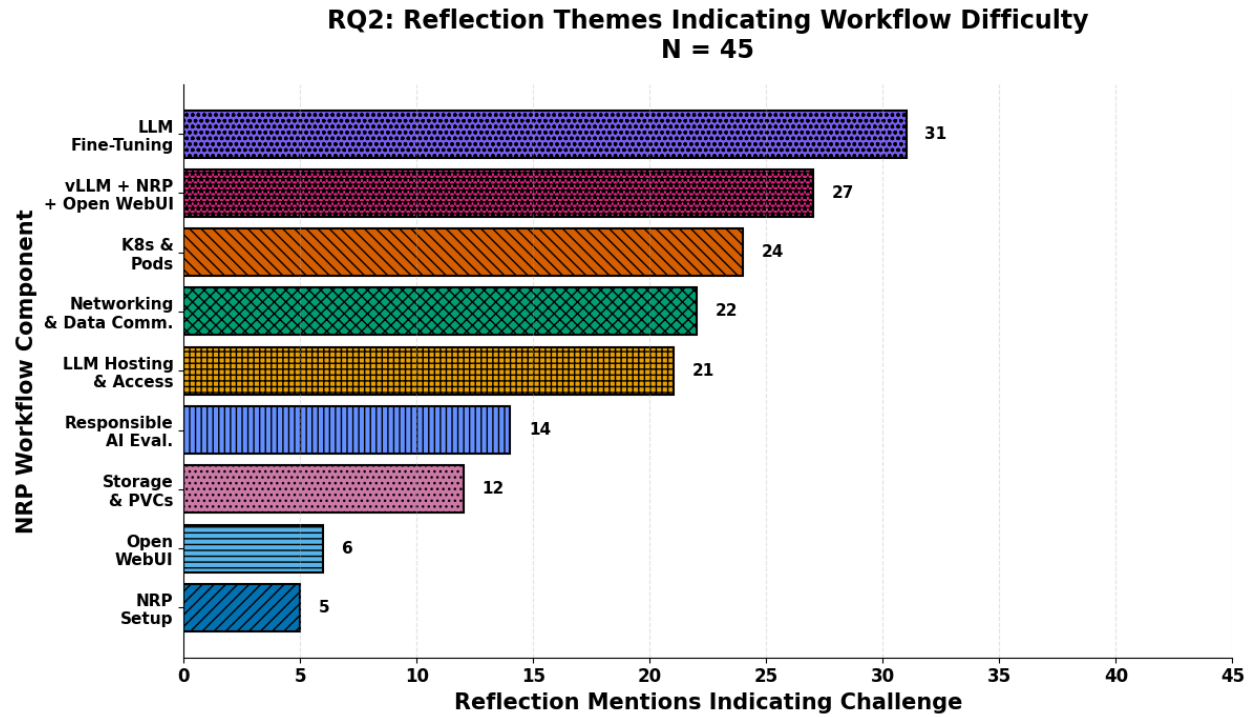


Figure 6. Reflection Themes Indicating Workflow Difficulty.

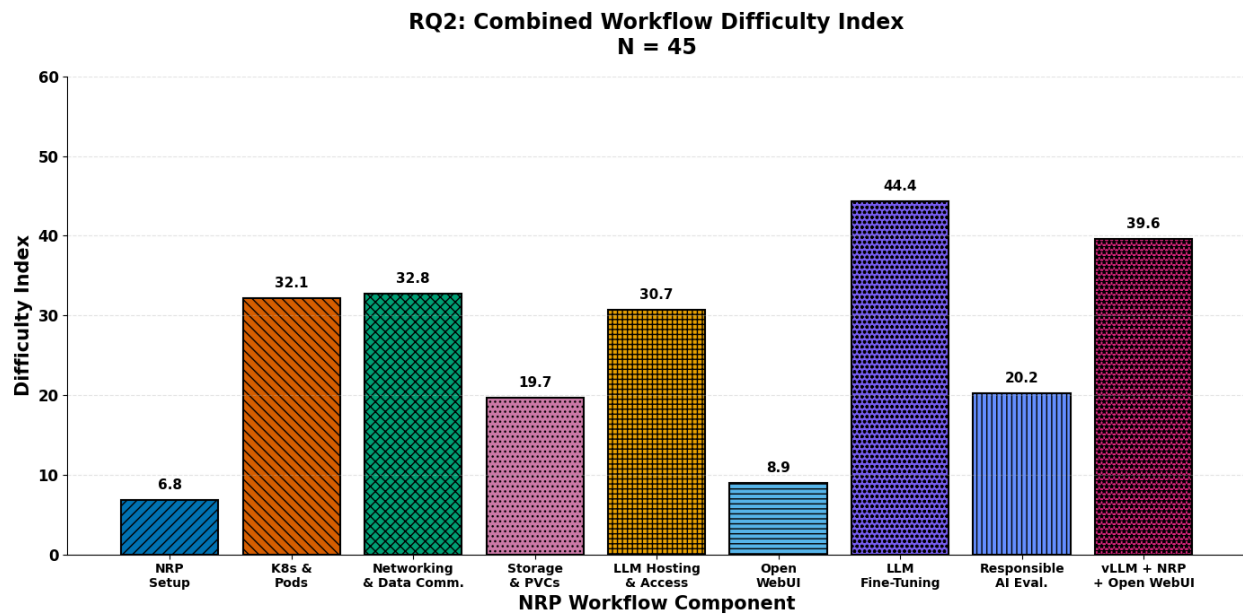


Figure 7. Combined Workflow Difficulty Index.

5.3 Competency Assessment (RQ3)

RQ3 examined the competency levels demonstrated across the NRP workflow domains. Section 3.4 discusses the metrics used for RQ3. Figure 8 presents the competency distributions across the nine workflow domains (Table E1 in Appendix E shows the mapping between competency domain and modules).

Figure 8 shows that NRP access and Open WebUI interaction produced the strongest competency outcomes. The distributions for these domains concentrate near the proficient and advanced levels, indicating that the cohort successfully accessed the NRP environment, navigated the workflow, interacted with hosted open-source LLMs, constructed prompts, and documented model outputs. These results indicate that the user-facing and access-oriented components of the workflow were readily adopted.

Responsible AI assessment and storage also produced strong competency outcomes. The cohort demonstrated the ability to evaluate model outputs, identify hallucination risks, discuss fairness concerns, and recognize limitations in generated responses. The storage domain also showed relatively strong performance, indicating that participants could manage datasets and outputs within the workflow.

Kubernetes, pod management, networking, and LLM hosting demonstrated moderate competency levels with broader distributions. These results indicate that the cohort could complete infrastructure-oriented activities but showed more variation when interpreting YAML files, requesting compute resources, explaining services and endpoints, and understanding model-serving architecture. The broader distributions suggest that these domains required more instructional support than access-oriented activities.

LLM adaptation and fine-tuning produced the widest distribution and the lowest overall competency profile. Although the cohort completed the adaptation workflow, participants demonstrated varying levels of success when modifying training examples, interpreting adaptation parameters, comparing model outputs, and explaining model behavior after adaptation. This finding indicates that model adaptation represented the most technically demanding component of the NRP workflow.

These findings demonstrate that the workflow enabled practical competency development across infrastructure, model access, and responsible AI evaluation while identifying advanced model adaptation as the area most in need of future instructional refinement.

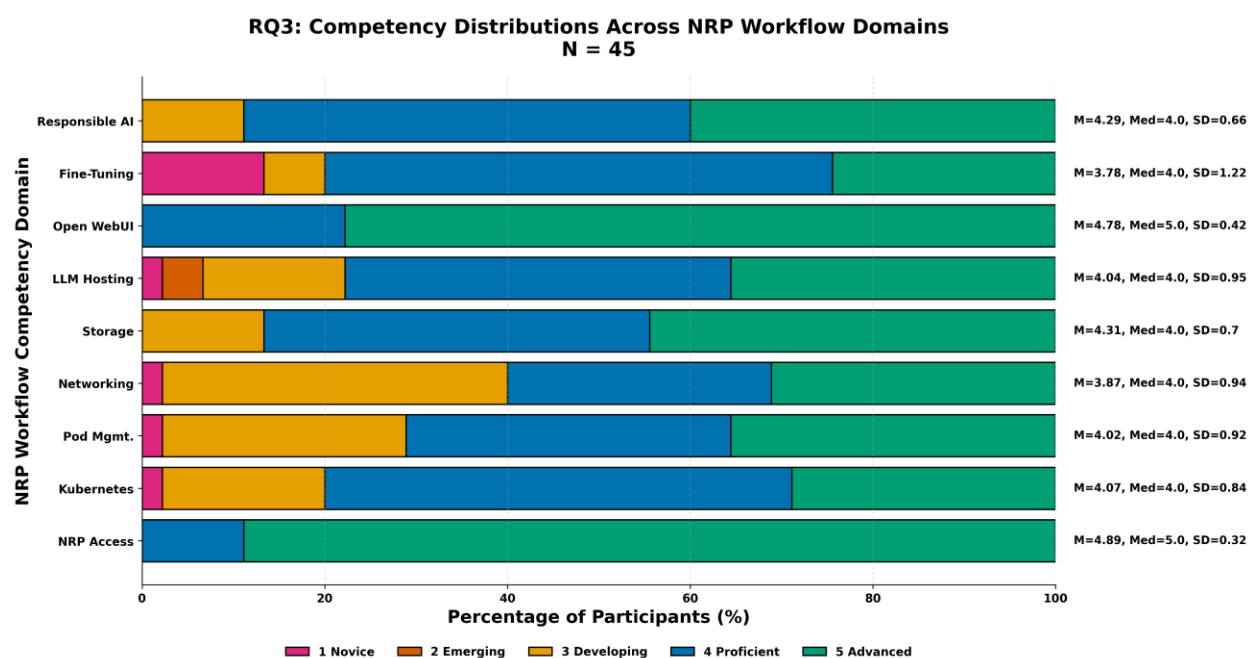


Figure 8. Competency Distributions Across NRP Workflow Domains.

5.4 Capstone Experiment Outcomes (RQ4)

RQ4 examines how effectively the cohort integrated complex cloud-native infrastructure concepts and

LLM workflows within the final capstone experiment. Figure 9 displays these integration outcomes across all nine individual rubric domains. Unlike the isolated competency distributions reported in RQ3, this visualization tracks collective deployment readiness and operational convergence across the complete engineering pipeline. All 45 students were evaluated in this capstone rubric; the 7 students who did not successfully complete Lab 9 (as reported in Figure 4) represent the individuals categorized as "Incomplete / Non-Functional" across the advanced integration domains.

The empirical findings indicate that the cohort successfully synthesized multiple disparate layers of the infrastructure stack, progressing beyond isolated laboratory activities to demonstrate full pipeline command. As shown in Figure 9, the cohort achieved maximum operational convergence in foundational environment provisioning and front-end interface binding. Both Domain 1 (NRP Access) and Domain 7 (Open WebUI Integration) yielded a 100% success rate executing their tasks completely independently. Foundational infrastructure tasks also showed high success, with Domain 2 (Kubernetes Concepts) and Domain 3 (Pod Management) yielding 40 and 42 fully operational independent deployments, respectively. Similarly, the cohort demonstrated robust engineering execution in Domain 5 (Storage Management) with 42 participants achieving independent operational status, and Domain 6 (LLM Hosting) where 40 participants successfully deployed functional containerized vLLM serving endpoints. The evaluation metrics also validate the cohort's performance in algorithmic governance, as 43 participants independently executed prompt validation matrices and safety audits within Domain 9 (Responsible AI Evaluation).

Conversely, the data isolates distinct technical friction points within the networking and parameter-adaptation layers of the pipeline. Domain 4 (Networking and Data Communication) produced the greatest performance variation across the entire study. While 28 participants successfully exposed their endpoints independently and 10 participants achieved operational status with instructional support, 7 participants produced incomplete or non-functional configurations, failing to resolve communication disconnects between the backend pods and the Open WebUI layer. These bottlenecks directly mirror the microservice routing challenges identified under RQ2.

Furthermore, Domain 8 (Model Adaptation) represented the most intellectually demanding integration segment. Although 37 participants independently succeeded in mounting their custom LoRA and QLoRA weights into the active container pipeline, a subset of 8 participants produced incomplete or non-functional artifacts, failing to successfully integrate their adapted models. Participants who mastered this adaptation workflow produced highly comprehensive qualitative analyses of generative performance. Ultimately, the final capstone experiment demonstrates that the cohort achieved successful systemic integration, proving the educational viability of utilizing decentralized, high-performance infrastructures to build scalable, end-to-end generative AI engineering competencies.

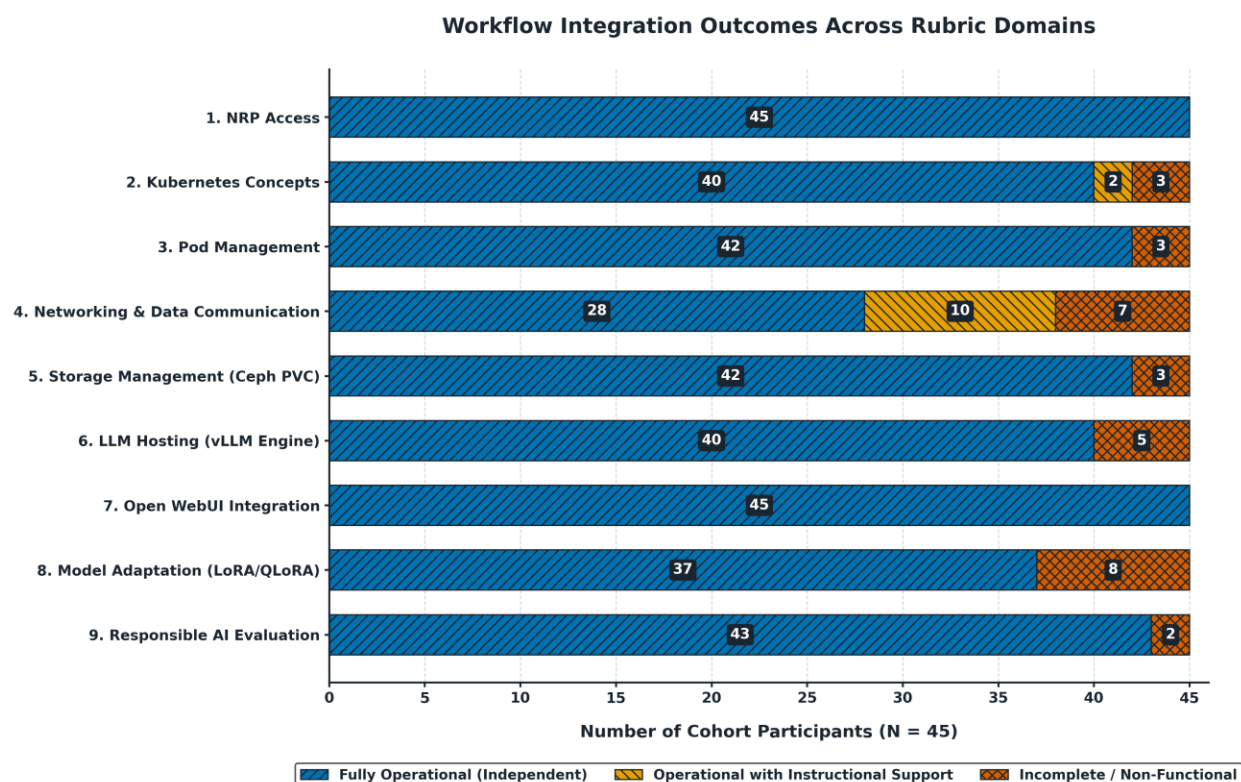


Figure 9. Workflow Integration Outcomes Across Rubric Domains.

6. DISCUSSION

The findings show that the NRP-based cloud-native AI workflow provided a feasible structure for introducing students to operational LLM infrastructure. The cohort completed most modules at high rates and reported positive perceptions of the technology-supported laboratory sequence. Participants completed access-oriented and user-facing tasks, such as NRP setup and Open WebUI interaction, with the strongest outcomes, while infrastructure-heavy activities involving Kubernetes, networking, vLLM integration, and model adaptation produced greater variation. This pattern suggests that the scaffolded sequence helped participants enter the workflow successfully, but the more complex infrastructure layers required additional instructional support. The course evaluation results further indicate that participants viewed the laboratory structure as useful for developing technology skills and understanding course concepts, although these post-course data should be interpreted as perception and feasibility evidence rather than causal evidence of learning gains.

The final capstone experiment showed that students could synthesize multiple technical layers, while also revealing persistent challenges in service routing, endpoint exposure, and model adaptation. These findings support the need for additional scaffolding around YAML configuration, Kubernetes networking, GPU-backed model serving, and interpretation of fine-tuning outcomes. Overall, the study contributes an applied model for cloud-native AI infrastructure education by demonstrating how shared research cyberinfrastructure can support practical LLM deployment, experimentation, and evaluation within technology-adjacent STEM courses.

7. LIMITATIONS & FUTURE WORK

This study carries several limitations: (1) The instructor served as the sole evaluator, introducing potential rater bias; (2) The lack of a control group prevents isolating the NRP framework's specific causal impact; (3) The absence of a quantitative pre-measure means results reflect perceived competency and qualitative confidence shifts rather than absolute gains; (4) The 45-student cohort limits immediate generalizability; and (5) The curriculum introduces substantial instructional overhead

to maintain evolving open-source libraries. Future work will integrate validated pre- and post-module self-efficacy surveys, expand to multi-tiered cohorts across other institutions, and compare the NRP framework against commercial alternatives like AWS and GCP. (A comprehensive discussion of study limitations and planned future research extensions is provided in Appendix I).

8. CONCLUSION

This paper introduces a scalable educational framework comprising nine specialized modules that train a cohort to leverage the NRP for deploying open-source LLMs and mastering hands-on generative AI skills. Evaluation data indicate that the cohort highly valued the curriculum, and qualitative Discord reflections show that the hands-on NRP workflow and Open WebUI integration provided an engaging experience, fostering self-reported increases in students' confidence to utilize LLMs for real-world use cases. The primary novelty of this work lies in its architectural and pedagogical blueprint, which fills a critical gap in equitable AI education by providing a concrete methodology to leverage decentralized computing power. This proposed pedagogical framework effectively lowers financial barriers for diverse institutions, offering IS and broader STEM instructors an adaptable model for infusing advanced computing concepts into analytical curricula.

9. REFERENCES

- ABET. (2026). Criteria for accrediting engineering programs, 2026–2027. <https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2026-2027/>
- Adams, J., Hainey, B., White, L., Foster, D., Hall, N., Hills, M., Hooshangi, S., Kuber, K., Nazir, S., Sakr, M., Stott, L., & Taglienti, C. (2020). Cloud computing curriculum: Developing exemplar modules for general course inclusion. In ITiCSE-WGR '20: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (pp. 151–172). Association for Computing Machinery. <https://doi.org/10.1145/3437800.3439206>
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191–215. <https://doi.org/10.1037/0033-295X.84.2.191>
- Breivold, H. P., & Crnkovic, I. (2014). Cloud computing education strategies. In 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEET) (pp. 29–38). IEEE. <https://doi.org/10.1109/CSEET.2014.6816778>
- Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient fine-tuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.52202/075280-0441>
- Foster, D., White, L., Adams, J., Erdil, D. C., Hyman, H., Kurkovsky, S., Sakr, M., & Stott, L. (2018). Cloud computing: Developing contemporary computer science curriculum for a cloud-first future. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 130–147). Association for Computing Machinery. <https://doi.org/10.1145/3293881.3295781>
- Govea, J., Ocampo Edye, E., Revelo-Tapia, S., & Villegas-Ch., W. (2023). Optimization and scalability of educational platforms: Integration of artificial intelligence and cloud computing. *Computers*, 12(11), Article 223. <https://doi.org/10.3390/computers12110223>
- Grafana Labs. (2026). Technical documentation. Grafana Documentation. <https://grafana.com/docs/>
- Han, Z., Gao, C., Liu, J., Zhang, J., & Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv*. <https://arxiv.org/abs/2403.14608>
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *Tenth International Conference on Learning Representations*. <https://openreview.net/forum?id=nZeVKeeFYf9>

- Kang, L., & Zhu, L. (2025). Design and application of digital teaching resource platform based on cloud-learning framework and deep learning algorithms. In 2025 Third International Conference on Networks, Multimedia and Information Technology (NMITCON) (pp. 1–7). IEEE. <https://doi.org/10.1109/NMITCON65824.2025.11188379>
- Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice Hall.
- Kubernetes. (2026a). kubelet. Kubernetes Documentation. <https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/>
- Kubernetes. (2026b). The kubectl command-line tool. Kubernetes Documentation. <https://kubernetes.io/docs/concepts/overview/kubectl/>
- Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. In Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP '23). <https://doi.org/10.1145/3600006.3613165>
- Leidig, P., & Salmela, H. (2021). IS2020: A competency model for undergraduate programs in information systems. Association for Computing Machinery and Association for Information Systems. <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is2020.pdf>
- Lens Documentation. (2026). Lens documentation. <https://docs.k8slens.dev/>
- Li, Q. (2021). The use of artificial intelligence combined with cloud computing in the design of education information management platform. *International Journal of Emerging Technologies in Learning*, 16(5), 32–44. <https://doi.org/10.3991/ijet.v16i05.20309>
- Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large language models: A survey. *arXiv*. <https://arxiv.org/abs/2402.06196>
- Ojika, F. U., Owobu, W. O., Abieba, O. A., Esan, O. J., Ubamadu, B. C., & Daraojimba, A. I. (2023). Transforming cloud computing education: Leveraging AI and data science for enhanced access and collaboration in academic environments. *Journal of Frontiers in Multidisciplinary Research*, 4(1), 138–156. <https://doi.org/10.54660/IJFMR.2023.4.1.138-156>
- Qasem, Y. A. M., Abdullah, R., Jusoh, Y. Y., Atan, R., & Asadi, S. (2019). Cloud computing adoption in higher education institutions: A systematic review. *IEEE Access*, 7, 63722–63744. <https://doi.org/10.1109/ACCESS.2019.2916234>
- Rad, P., Roopaei, M., Beebe, N., Shadaram, M., & Au, Y. A. (2018). AI thinking for cloud education platform with personalized learning. In Proceedings of the 51st Hawaii International Conference on System Sciences (pp. 1–10). <https://doi.org/10.24251/HICSS.2018.003>
- Rana, A., Vaidya, P., & Hu, Y.-C. (2026). Transformative role of large language models in education and comparative analysis with traditional technologies. *Education and Information Technologies*, 31, 1137–1162. <https://doi.org/10.1007/s10639-025-13854-5>
- Shi, Y., Yu, K., Dong, Y., & Chen, F. (2026). Large language models in education: A systematic review of empirical applications, benefits, and challenges. *Computers and Education: Artificial Intelligence*, 10, Article 100529. <https://doi.org/10.1016/j.caeai.2025.100529>
- Weil, S. A., Brandt, S. A., Miller, E. L., Long, D. D. E., & Maltzahn, C. (2006). Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)* (pp. 307–320). USENIX Association. <https://www.usenix.org/conference/osdi-06/ceph-scalable-high-performance-distributed-file-system>

- Weitzel, D., Graves, A., Albin, S., Zhu, H., Würthwein, F., Tatineni, M., Mishin, D., Graham, J., Khoda, E. E., Sada, M. F., Smarr, L., & DeFanti, T. (2025). The National Research Platform: Stretched, multi-tenant, scientific Kubernetes cluster. arXiv. <https://arxiv.org/abs/2505.22864>
- Weng, Q., Yang, L., Yu, Y., Wang, W., Tang, X., Yang, G., & Zhang, L. (2023). Beware of fragmentation: Scheduling GPU-sharing workloads with fragmentation gradient descent. In 2023 USENIX Annual Technical Conference (USENIX ATC 23) (pp. 995–1008). USENIX Association. <https://www.usenix.org/conference/atc23/presentation/weng>
- Yan, L., Sha, L., Zhao, L., Li, Y., Martínez-Maldonado, R., Chen, G., Li, X., Jin, Y., & Gašević, D. (2024). Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, 55(1), 90–112. <https://doi.org/10.1111/bjet.13370>

APPENDIX A

A.1 Cloud-Native Infrastructure for AI

Cloud-native infrastructure provides the operational foundation for modern AI systems by orchestrating containers, microservices, and declarative APIs. As an open-source platform, Kubernetes automates the deployment, scaling, and management of these containerized applications, abstracting workloads into distinct pods to guarantee scientific and operational reproducibility for complex workflows that rely on specific software dependencies and precise runtime parameters. Furthermore, Kubernetes enables precise resource management for intensive AI workloads by allowing users to explicitly specify required computing resources and accelerator types, while the Kubernetes agent (kubelet) (Kubernetes, 2026a) enforces strict resource quotas per namespace. This structured approach directly exposes students to essential cloud-native concepts such as managing namespaces, pods, services, ingress rules, and persistent storage volumes to deploy high-performance inference engines and integrate them with platform-hosted user interfaces.

The management and scheduling of GPUs remain a premier operational challenge within these clusters due to severe fragmentation risks. Weng et al. (2023) determined that static or partial GPU allocation frequently leaves substantial computing capacity unusable, proposing a Fragmentation Gradient Descent scheduler for Kubernetes that reduced unallocated cluster capacity by up to 49% using real-world production traces. Resolving these inefficiencies through auto-scaling, proactive job preemption, workload-aware scheduling, and elastic GPU sharing is an essential requirement for balancing concurrent model training and inference. Understanding these mechanics matters immensely because platforms like the NRP allow students to transition from theoretical abstractions to real-world infrastructure management. By deploying high-performance inference engines such as vLLM (Kwon et al., 2023) and directly monitoring cluster performance, students learn to optimize resource allocation and handle the severe hardware demands inherent in distributed AI pipelines.

A.2 Cloud Computing and AI Infrastructure Education

While traditional cloud computing curricula emphasize using cloud technologies as external, turnkey solutions, industry demand requires graduates who can actively engineer, manage, and scale these architectures (Foster et al., 2018; Adams et al., 2020). Scholars have developed various structural frameworks to address this need, including dedicated cloud education management platforms (Li, 2021), virtual laboratories that synthesize data science and cloud principles (Ojika et al., 2023), and deep learning delivery platforms that optimize student engagement (Kang & Zhu, 2025). However, educational institutions face systemic barriers such as security, privacy, and commercial vendor lock-in (Qasem et al., 2019), which highlights the necessity of a broad coverage of underlying cloud architectures (Breivold & Crnkovic, 2014) using cost-effective alternatives. This pedagogical gap is further widened by the current state of AI education; despite an exponential growth in literature, the vast majority of studies focus on how learners interact with LLMs as passive end-users via chatbots or automated tutors, which introduces risks regarding low technological readiness, student overreliance, and reduced critical thinking (Rana et al., 2026; Shi et al., 2026; Yan et al., 2024). To shift students from passive consumers to active developers, emerging frameworks focus on the programmatic convergence of computational resources, integrating AI models with cloud backends to optimize personalization (Govea et al., 2023) and utilizing interactive virtual spaces like Cloud-eLab to let students independently train models (Rad et al., 2018). This study builds upon these precedents by training the cohort to engineer the containerized infrastructure, network routing rules, and distributed storage systems required to serve and fine-tune open-source models rather than merely utilizing AI as an instructional aid.

APPENDIX B

Detailed NRP Workflow Architecture

As outlined in Figure 2, the NRP architecture organized the course environment around Kubernetes-managed resources within a dedicated namespace. Pods supported the GPU-based workloads, encapsulating the containers and specifying hardware resource requirements such as CPU, GPU, and RAM. Participants utilized tools such as kubectl, OpenLens, the NRP portal, and Grafana to inspect these pods, monitor execution status, view hardware utilization metrics, and troubleshoot deployment bottlenecks.

The service layer securely connected these backend AI workloads to user-facing applications. Kubernetes services provided stable network routes that abstracted the temporary IP addresses of individual backend pods, allowing the vLLM inference server to communicate directly with Open WebUI through an OpenAI-compatible API endpoint. This structural separation of the frontend interaction layer from the backend inference layer mirrored production enterprise environments.

For the model-serving layer, the architecture relied on containerized vLLM inference services to run both base open-source and fine-tuned LLMs on the assigned GPU hardware. The vLLM server exposed these models via REST API endpoints, supporting efficient inference through continuous batching and optimized key-value cache management.

Finally, Open WebUI operated as the primary user-facing interface. Through this browser-based environment, participants selected active models, submitted prompts, reviewed generation histories, and interacted with customized LLM applications without relying exclusively on command-line interactions, preserving accessibility while maintaining a direct connection to the underlying Kubernetes infrastructure.

APPENDIX C

Assessment Area	Evaluation Focus and Criteria	Evidence Measured	Theoretical Alignment
Completion	Successful execution of sequential laboratory procedures and workflow milestones	System submission timestamps and repository records	Supports Bandura's self-efficacy theory by capturing mastery experiences through successful task completion. Also reflects Kolb's concrete experience stage through direct engagement with NRP, Kubernetes, Open WebUI, and LLM workflows.
Performance	Academic accuracy, correctness of configurations, and problem-solving quality across the nine modules	Structured assignment grades, configuration checks, code validation, and instructor review	Supports Bandura by measuring successful performance under increasing technical complexity. Supports Kolb's reflective observation and abstract conceptualization stages as the cohort interprets errors, outputs, and system behavior.
Competency	Demonstrated mastery across discrete cloud-native and AI workflow domains, including Kubernetes, pod management, networking, storage, LLM hosting, Open WebUI, LLM adaptation, and responsible AI assessment	Normalized rubric scores applied to laboratory artifacts and workflow deliverables	Supports Bandura by evaluating confidence-building mastery across technical domains. Supports Kolb by showing progression from hands-on experience to conceptual understanding and applied capability.

Integration	Independent synthesis of prior infrastructure and modeling concepts during the final capstone project	Final vLLM, NRP, Open WebUI, and fine-tuned model deployment artifacts, endpoint configuration, ingress rules, and validation outputs	Supports Bandura by evaluating whether the cohort can persist through and complete a complex applied task. Supports Kolb's active experimentation stage through independent integration, testing, refinement, and deployment.
-------------	---	---	---

Table C1. Assessment Framework and Theoretical Alignment.

APPENDIX D

Extended Analysis Strategy and Metric Calculations

To rigorously address the RQs introduced in this paper, we map specific quantitative and qualitative metrics to each inquiry.

Our strategy for RQ1 measures overall student engagement and system viability by analyzing aggregate laboratory completion rates across the scaffolded workflow.

For RQ2, we evaluate workflow execution difficulties and performance variations. We analyze quantitative assignment outcomes across all nine scaffolded laboratories alongside a qualitative thematic analysis of individual student reflection responses to isolate recurring execution challenges. To synthesize objective performance and self-reported difficulty into a single comparative measure, a Combined Workflow Difficulty Index was calculated for each of the nine workflow components, weighting inverse assignment performance (60%) and reflection-cited difficulty (40%); the full formula is reported in Section 5.

For RQ3, we assess competency distributions across nine core cloud-native infrastructure domains: NRP Access, Kubernetes Fundamentals, Pod and Resource Management, Networking and Data Communication, Storage Management, LLM Hosting, Open WebUI Integration, Model Adaptation (LoRA/QLoRA), and Responsible AI Evaluation. These domains reflect competency built cumulatively across Modules 1 through 8 described in Section 4; they are distinct from the modules themselves, and their relationship is detailed in Table B1 in Appendix B. We normalize rubric-based evaluations onto a continuous 0–5 competency index, calculated as $(\text{Rubric Points Earned} / \text{Maximum Rubric Points}) \times 5$. Higher values indicate stronger demonstrated mastery; for interpretive purposes, scores can be grouped into approximate bands (e.g., <2 = novice/emerging, $2-3$ = developing, $3-4$ = proficient, $4-5$ = advanced). We calculate this metric using the formula:

$$\text{Competency Score} = (\text{Rubric Points Earned} / \text{Maximum Rubric Points}) \times 5 \quad (1)$$

This standardization allows for a consistent evaluation across all domains, irrespective of varying raw point values between assignments.

Difficulty index formula is calculated using the formula:

$$\text{Difficulty Index} = [((100 - \text{Mean Assignment Score}) / 100) \times 0.6 + (\text{Reflection Mentions} / N) \times 0.4] \times 100 \quad (2)$$

Where Mean Assignment Score is the component's average score (%), Reflection Mentions is the count of student reflections citing that component as a source of difficulty, and $N = 45$. The index weights assignment performance more heavily than reflection frequency (60% vs. 40%) because objective performance data are available for all participants, while reflection mentions depend on which topics students chose to write about and are therefore a noisier, self-selected signal of difficulty. The index has a theoretical range of 0 (highest possible performance, zero reflection mentions) to 100 (lowest

possible performance, every participant citing the component); observed values in this study ranged from 6.84 (NRP Setup) to 44.36 (LLM Fine-Tuning).

Finally, for RQ4, we assessed the students capability to synthesize these disparate technical layers by evaluating the empirical outcomes of the final capstone experiment. This comprehensive lab requires all students to orchestrate a live, end-to-end generative AI engineering pipeline within an operational NRP environment. To evaluate this multi-layer synthesis, the assessment framework aggregates individual student rubric outcomes across all nine distinct operational infrastructure domains discussed above. We normalize these individual domain metrics to derive a final composite project score using a standardized five-point competency index. This comprehensive score allows us to evaluate holistic system integration and deployment readiness rather than isolated technical skills.

APPENDIX E

IS2020 Alignment of NRP-Based Cloud-Native AI Modules

Module	Module Focus	Relevant IS2020 Competency Areas	Alignment Rationale	Competency Domain
Module 1: NRP Onboarding and Infrastructure Fundamentals	NRP access, namespace setup, platform orientation, resource awareness	Foundations of Information Systems; IT Infrastructure; Emerging Technologies	Introduces the role of shared cyberinfrastructure in supporting modern IS applications and exposes participants to contemporary cloud-native AI platforms.	NRP Access
Module 2: Kubernetes and Pod Requests	Kubernetes concepts, pod deployment, resource requests, CPU/GPU/RAM allocation	IT Infrastructure; Application Development and Programming; Emerging Technologies	Develops operational understanding of containerized infrastructure, workload deployment, resource provisioning, and infrastructure-as-code concepts.	Kubernetes Fundamentals & Pod and Resource Management
Module 3: Networking and Data Communication	Services, endpoints, routing, API communication, pod-to-service interaction	IT Infrastructure; Web Development; Secure Computing	Aligns with networked system design by requiring participants to configure and validate service communication, routing, and endpoint accessibility.	Networking and Data Communication
Module 4: Storage and Persistent Volumes	Persistent volume claims, datasets, workflow artifacts, storage persistence	Data/Information Management; IT Infrastructure	Connects data management with infrastructure by requiring participants to manage persistent datasets, outputs, model artifacts, and reproducible workflow files.	Storage Management

Module 5: LLM Hosting and Inference Engine Orchestration	Open-source LLM hosting, GPU-backed inference, endpoint validation	IT Infrastructure; Emerging Technologies; Application Development and Programming	Supports competency in deploying AI-enabled services through containerized model-serving infrastructure and GPU-backed inference workflows.	LLM Hosting
Module 6: Open WebUI Workflow and Prompt Evaluation	Browser-based model interaction, prompt testing, model selection, response evaluation	User Interface Design; Web Development; Emerging Technologies	Aligns with user-facing IS design by connecting backend model-serving services to an accessible frontend interface for model interaction and evaluation.	Open WebUI Integration
Module 7: LLM Fine-Tuning and Parameter-Efficient Adaptation	SFT, LoRA/QLoRA, training examples, base vs. adapted model comparison	Application Development and Programming; Data/Business Analytics; Emerging Technologies	Develops applied AI customization skills by requiring participants to prepare data, adapt models, compare outputs, and interpret model behavior.	Model Adaptation (LoRA/QLoRA)
Module 8: Responsible AI Evaluation and Algorithmic Guardrails	Hallucination, bias, fairness, privacy, safety, model limitations	IS Ethics, Sustainability, Use and Implications for Society; Secure Computing; Data/Information Management	Aligns with ethical and secure IS practice by requiring participants to evaluate risks, document limitations, and propose guardrails for responsible deployment.	Responsible AI Evaluation
Module 9: vLLM Integration with NRP and Open WebUI	End-to-end integration of NRP, Kubernetes, vLLM, Open WebUI, fine-tuned models, and evaluation	IS Practicum; IS Project Management; IT Infrastructure; Emerging Technologies; Application Development and Programming	Serves as the culminating integration activity in which participants synthesize infrastructure, model-serving, frontend interaction, storage, and evaluation concepts into an operational workflow.	

Table E1. Alignment of NRP-based modules with IS2020 competency areas.

APPENDIX F

Open-Source Laboratory Modules and Instructional Artifacts

To support pedagogical adoption and ensure reproducibility, the complete open-source curriculum is publicly available via static web repositories. Each hosted environment contains the module's structural overview, explicit command-line execution steps, boilerplate YAML deployment manifests, configuration files, and participant exercise handouts. The following anonymous links provide full access to the instructional assets used throughout this study:

Module	Title	Anonymous Review Link
Module 1	NRP Onboarding and Infrastructure Fundamentals	https://silver-sprinkles-1c184e.netlify.app/
Module 2	Kubernetes Fundamentals and Containerized Resource Requests	https://polite-lamington-b3e830.netlify.app/
Module 3	Networking and Microservice Communication	https://celebrated-bienenstich-3a1469.netlify.app/
Module 4	Storage Infrastructure and Persistent Volumes	https://keen-paletas-00247e.netlify.app/
Module 5	LLM Hosting and Inference Engine Orchestration	https://snazzy-gelato-f1f171.netlify.app/
Module 6	Open WebUI Workflow and Prompt Engineering Evaluation	https://tourmaline-hummingbird-589cb6.netlify.app/
Module 7	LLM Fine-Tuning and Parameter-Efficient Adaptation	https://moonlit-horse-f6d241.netlify.app/
Module 8	Responsible AI Evaluation and Algorithmic Guardrails	https://luminous-vacherin-1bd441.netlify.app/
Module 9	vLLM Integration with NRP and Open WebUI	https://incandescent-kheer-77753.netlify.app/

Table F1. NRP-based LLM laboratory modules and review links.

APPENDIX G

Detailed Laboratory Execution Workflows

G.1 Module 1: NRP Onboarding and Infrastructure Fundamentals

The introductory module connects students to the core architecture of the NRP through account configuration, namespace registration, and initial resource provisioning. Students master foundational cloud-native infrastructure concepts required to host and run LLMs, focusing specifically on namespaces, pods, computing jobs, GPU resources, and persistent storage volumes. To complete the practical component, students register for an NRP account using their institutional email credentials. Because the NRP requires strict administrative grouping, students must join our dedicated department namespace before requesting computational allocations. Following namespace verification, students modify pre-configured YAML templates with their namespace identifiers to execute their first resource request. Students then actively monitor the deployment queue to observe cluster scheduling and provisioning. In the final phase, students execute Linux commands to inspect the container environment and validate that the active hardware specifications match their requested CPU, RAM, and GPU allocations. Additionally, this module teaches the core tenets of the NRP fair usage policy, training students to monitor namespaces for resource violations and terminate active deployments immediately upon project completion to preserve shared cluster capacity.

G.2 Module 2: Kubernetes Fundamentals and Containerized Resource Requests

This module introduces core principles of container orchestration using Kubernetes, requiring students to provision and manage pods, jobs, and deployments. Students analyze the computational requirements of their workloads to determine and request appropriate allocations of CPU, GPU, memory, and storage resources. A primary objective is to navigate the NRP documentation and workspace to request cluster resources accurately. The laboratory introduces the syntax and structure of YAML configuration files. Students author precise declarative files specifying hardware requirements and

software stacks, then submit them to the cluster via the command-line interface (CLI). Through this process, students interpret real-time deployment statuses and troubleshoot common scheduling bottlenecks.

G.3 Module 3: Networking and Microservice Communication

This module explores data communication within a Kubernetes cluster, focusing on how AI microservices interact. The integrated laboratory requires students to deploy a lightweight inference service, expose it via a Kubernetes service abstraction, and trace HTTP request payloads. This exercise models how modern enterprises deploy containerized microservices and manage internal traffic routing. Additionally, students address infrastructure security by configuring designated ports, deploying multiple decoupled pods, and establishing cross-pod communication using internal ClusterIP services. The first phase requires deploying an AI inference service backed by GPU accelerators, mapping container ports, configuring an Ingress controller, and analyzing the end-to-end HTTP packet flow.

G.4 Module 4: Storage Infrastructure and Persistent Volumes

This module explores distributed storage options within the NRP, training the students to configure and attach stable storage to active computing workloads. Because deep learning workflows require persistent storage to retain massive datasets and model weights across container lifecycles, participants learn to implement Persistent Volume Claims (PVCs). The underlying storage infrastructure utilizes a Ceph-backed distributed file system (Weil et al., 2006), which dynamically mounts storage volumes to any available cluster node on demand rather than tethering data to a single physical server. The laboratory requires each student to provision a dedicated PVC within their assigned namespace. Participants then integrate this PVC into their YAML configuration files to mount the persistent storage layer directly alongside their GPU resource allocations. Once the cluster provisions the environment, the students connect via the terminal to verify the directory structure and validate that the mount point functions correctly. In the final phase of the laboratory, each student builds an isolated Python virtual environment inside the persistent volume and downloads a foundational open-source LLM for use in subsequent modules.

G.5 Module 5: LLM Hosting and Inference Engine Orchestration

This module covers the core mechanics of LLMs, demonstrating how critical hyperparameters like temperature and Top-K sampling alter text generation. In this module, students utilize the high-performance inference engine vLLM to host, configure, and serve lightweight open-source architectures (e.g., Llama-3-8B or Qwen2.5-0.5B-Instruct) on the NRP infrastructure. During the laboratory component, participants stream model weights directly from Hugging Face into their persistent storage volumes, launch the model-serving endpoint, and issue programmatic curl queries to evaluate real-time responses. Finally, the students modify context lengths and serving parameters to gather empirical performance data, allowing them to determine optimal configuration metrics for diverse operational scenarios.

G.6 Module 6: Open WebUI Workflow and Prompt Engineering Evaluation

This module introduces Open WebUI, a user-facing containerized interface that connects users directly to underlying inference endpoints. Students access the institutional Open WebUI portal using their NRP credentials and generates individualized API tokens to interact with hosted models programmatically via OpenAI-compatible endpoints. Once configured, participants select an open-source LLM from the cluster and execute structured and unstructured prompt templates across varying temperature settings to evaluate changes in text generation. Finally, students use a standardized rubric to systematically evaluate output quality, formatting adherence, and structural precision, establishing which prompting strategies yield the highest operational efficacy.

G.7 Module 7: LLM Fine-Tuning and Parameter-Efficient Adaptation

This module covers the principles of instruction fine-tuning, demonstrating how developers adapt a base LLM using specialized instruction-response datasets to inject domain-specific knowledge and alter model behavior. The first component of the laboratory requires the students to provision an NRP computing

environment equipped with GPUs and a Jupyter Notebook interface to manage the execution of the training pipeline. The cohort selects a base instructional model, such as Qwen2.5-0.5B-Instruct, along with a targeted training dataset. Using these prerequisites, the cohort executes a Supervised Fine-Tuning (SFT) workflow leveraging Parameter-Efficient Fine-Tuning (PEFT) techniques, specifically Low-Rank Adaptation (LoRA) and Quantized Low-Rank Adaptation (QLoRA). This optimization process compresses training overhead, allowing the model to adapt within minutes. In the final phase, the cohort conducts a comparative evaluation by submitting identical prompts to both the original base model and their newly adapted fine-tuned model. The students modify the underlying tuning dataset, adjust learning hyperparameters within the SFT framework, re-run the optimization training pipeline, and document how these adjustments bring the model outputs closer to the target ground-truth data.

G.8 Module 8: Responsible AI Evaluation and Algorithmic Guardrails

This module guides students through evaluating LLM outputs for critical vulnerabilities, including hallucinations, fairness, systematic bias, data privacy, and compliance governance. The laboratory component first requires the students to compile a specialized evaluation dataset engineered to intentionally probe and elicit these undesirable model behaviors. Next, the students connect to the NRP Open WebUI layer to query open-source models under low-temperature configurations using the evaluation dataset. The cohort deploys automated evaluation scripts to calculate baseline performance metrics while simultaneously conducting human-in-the-loop qualitative evaluations to judge fairness, contextual hallucinations, and compliance alignment. Ultimately, students synthesize these automated metrics and qualitative judgments into a formal assessment report. Based on their structural findings, the students propose specific guardrail implementations and explicitly document the operational limitations of the evaluated models.

G.9 Module 9: vLLM Integration with NRP and Open WebUI

This final lab serves as the final capstone experiment for the NRP workflow. Unlike earlier modules that focused on isolated technical components, this lab requires the cohort to synthesize all core curriculum concepts, integrating cloud-native infrastructure, LLM deployment, model serving, Open WebUI configuration, and model evaluation into a single end-to-end pipeline. The laboratory begins with vLLM deployment on the NRP Kubernetes environment. The cohort deploys a vLLM instance hosting an open-source LLM, verifies that the model-serving endpoint is operational, configures the required API endpoints, and validates connectivity from external clients. Next, the cohort configures Kubernetes ingress resources to expose the model-serving endpoint, applying networking concepts from previous modules. The cohort verifies that the ingress configuration correctly routes traffic to the model-serving pod and that external requests reach the deployed LLM. Once the endpoint is accessible, the cohort connects the vLLM instance to the NRP-hosted Open WebUI. Participants log into Open WebUI with administrative privileges, register the vLLM endpoint, configure the API key, and validate communication between the user interface and the model. This stage allows the cohort to interact with their deployed LLM through a graphical user interface. The cohort then evaluates the deployed model by executing multiple prompts through both Open WebUI and command-line interfaces. Participants compare model outputs, evaluate response quality, measure latency, and examine the impact of different temperature settings on model behavior. In the final stage, participants integrate the fine-tuned models they created in the previous lab. Students whose model artifacts are unavailable use an instructor-provided fine-tuned model to complete the workflow. The cohort deploys their adapted models, configures ingress resources, connects them to Open WebUI, and verifies successful interaction. Finally, the cohort compares outputs generated by the base model and the adapted model to evaluate whether the adaptation process produced measurable changes in model behavior. Because this comprehensive lab tests all key concepts, Lab 9 serves as the primary assessment artifact for RQ4.

APPENDIX H

Item	Scale	Valid N	Mean (SD)	Positive Response	P value, one-sided
-------------	--------------	--------------------	----------------------	------------------------------	-------------------------------

The instructor uses a variety of technological tools to enhance the course instruction.	5-point	38	4.45 (0.83)	84.2%	<.001***
The course information further developed my knowledge in this area.	5-point	39	4.49 (0.82)	84.6%	<.001***
The course activities enhanced my learning the course content.	5-point	39	4.49 (0.79)	87.2%	<.001***
The course activities improved my computer technology skills.	5-point	39	4.56 (0.75)	89.7%	<.001***
Lab or clinical experiences contributed to my understanding of the course theories and concepts.	5-point	34	4.53 (0.79)	88.2%	<.001***
The instructor clearly explained laboratory or clinical experiments or procedures.	5-point	33	4.45 (0.79)	87.9%	<.001***
My overall satisfaction with the instructor's teaching performance was.	4-point	39	3.51 (0.79)	87.2%	<.001***
My overall satisfaction with the course was.	4-point	39	3.51 (0.64)	92.3%	<.001***

Table H1. Student Survey Outcomes.

APPENDIX I

Extended Limitations and Future Work

This study carries several limitations. First, the instructor who delivered the curriculum also served as the sole evaluator who assigned every rubric score, completion judgment, and competency rating reported in Sections 5.1 through 5.4; we used no independent or blinded scoring, so we cannot rule out rater bias favoring the instructional design. Second, the study lacks a comparison group taught through a conventional, non-scaffolded approach, so we cannot attribute the observed outcomes specifically to the NRP-based workflow rather than to other factors such as instructor attention or student self-selection into the course. Third, while Bandura's self-efficacy was operationalized iteratively through qualitative student reflections posted to Discord following each module assessment, the study lacks a formalized quantitative pre-measure of student competency or self-efficacy. Because we cannot establish statistical gains relative to a baseline, readers should interpret the results as evidence of feasibility, perceived competency, and qualitative confidence shifts rather than of absolute, quantifiable learning gains. Fourth, the small pilot cohort of 45 students limits immediate generalizability, though these initial insights establish a foundational baseline for future scaling. Fifth, the curriculum introduces substantial instructional overhead: instructors must provide intensive guidance during early laboratory sessions to mitigate student friction, and they must continuously update scripts to maintain technical compatibility with rapidly evolving open-source software libraries. Finally, students experienced distinct initial difficulties conceptualizing the underlying NRP infrastructure, a challenge that diminished through sequential reinforcement but underscores the necessity of structured, scaffolded instructional support during the early phases of the curriculum.

Future work will extend this study during the Fall 2026 and Spring 2027 academic semesters by introducing updated laboratory exercises that directly address these pedagogical and technical challenges. To address the current evaluation constraints, this extension will integrate a validated, quantitative pre- and post-module self-efficacy survey alongside the continuous Discord reflections to systematically track and measure absolute competency shifts. The authors plan to scale these modules across other Information Systems, Data Analytics, and broader STEM disciplines. To evaluate the platform-agnostic effectiveness of this approach, upcoming research will compare the educational effectiveness of the NRP framework with commercial alternatives such as Google Cloud Platform and Amazon Web Services.

The authors also plan to collaborate with an expanded network of peer higher education and K-12 institutions to evaluate the framework across a larger, multi-tiered cohort. This expansion directly addresses the widening AI skills shortage in underserved educational sectors by broadening access to high-performance computing resources. Finally, enriching the curriculum will involve designing new instructional modules focused on Retrieval-Augmented Generation and agentic AI workflows.