

Generative-AI Assisted Feedback Provisioning for Project-Based Learning in CS Courses

Venkata Alekhya Kusam, University of Michigan, Dearborn

Venkata Alekhya Kusam is currently pursuing a Master's degree in Computer and Information Science at the University of Michigan-Dearborn. She has always been fascinated by the transformative power of technology. Her research interests lie in generative AI, large language models, and natural language processing (NLP).

Larnell Moore, University of Michigan, Dearborn

Larnell Moore is an undergraduate student in his final year pursuing a Bachelor's degree in Computer and Information Science at the University of Michigan-Dearborn. Alongside his studies, he has been working as a software engineer intern at the CSAA Insurance Group for a year. He has also worked as a supplemental instruction leader at the University of Michigan-Dearborn for two years. Larnell has been the recipient of numerous accolades such as the Rackham Merit Fellowship REA award, the title for the most exciting pitch at the 2023 CSAA Insurance Group Innovation Jam, the Destination Blue Scholarship, Slosberg and Sorscher Memorial Scholarship, CECS Richard Schaum Scholarship, funding for research under the NSF REU grant, and more. Post-graduation, Larnell Moore is set to pursue a Ph.D. in Computer Science and Engineering at the University of Michigan-Ann Arbor, aiming to further his research in Natural Language Processing, Deep Learning, and Machine Learning.

Summit Shrestha, University of Michigan, Dearborn

Summit Shrestha received his Bachelors of Science degree in Computer Science from Tribhuvan University in 2018. He is currently pursuing his Master's in Computer and Information Science at the University of Michigan-Dearborn, where he is also working as a Research Assistant with the Pervasive Computing Lab. Before joining the Master's program, he worked in software and systems as a Software Engineer, Lead Software Engineer, and then Principal Engineer with non-profit organizations and software companies overseas. His past experience working with educational and learning systems for K-12 education in rural Nepal has shaped his research focus on building system support for resource-constrained environments. His current research interests include edge computing and distributed systems, adaptive systems, middleware, sustainable and scalable edge systems, educational technologies and learning sciences.

Zheng Song, University of Michigan, Dearborn

Dr. Song received his second PhD in Computer Science (with a focus on distributed systems and software engineering) from Virginia Tech USA in 2020, and the first PhD (with a focus on wireless networking and mobile computing) from Beijing University of Posts and Telecommunications China in 2015. He worked as a software engineer at Sina for one year after I graduated as a master from China Agriculture University in 2009. He received the Best Paper Award from IEEE Edge in 2019.

Jin Lu, University of Georgia

Jin Lu received his Ph.D. degree in computer science and engineering from the University of Connecticut, USA in 2019. He worked as an assistant professor at the University of Michigan - Dearborn from 2019 to 2023. He is currently an assistant professor at the School of Computing at the University of Georgia. My major research interests include machine learning, data mining, and optimization. I am particularly interested in transparent machine learning models, distributed learning algorithms, optimization and so on.

Qiang Zhu, University of Michigan, Dearborn

Qiang Zhu received his Ph.D. degree in computer science from the University of Waterloo, Canada, in 1995. He is currently a Professor and the Chair of the Department of Computer and Information Science,



University of Michigan - Dearborn, USA. He has been honored as the William E. Stirton Professor at his University, an ACM Distinguished Scientist, an IEEE Senior Member, and an IBM CAS Faculty Fellow. His current research interests include query optimization for advanced database systems, big data processing and analytics, streaming data processing, spatio-temporal data processing, data mining, federated learning, and data security and privacy.

Generative-AI Assisted Feedback Provisioning for Project-based Learning in CS Courses

Venkata Alekhya Kusam¹*, Larnell Moore¹*, Summit Shrestha¹*, Zheng Song¹, Jin Lu², Qiang Zhu¹

{alekhyak, larnell, summitsh, zhesong}@umich.edu, jin.lu@uga.edu, qzhu@umich.edu

* The authors contribute equally to this work

 1 Dept. of Computer and Information Science, University of Michigan at Dearborn

² School of Computing, University of Georgia

Abstract

Project-Based Learning (PBL) is a pedagogical method that combines theory and practice by involving students in real-world challenges. Continuous feedback is crucial in PBL, guiding students to improve their methods and foster progressive thinking. However, PBL faces challenges in widespread adoption due to the time and expertise needed for effective feedback, especially with increasing student numbers. This paper presents a novel approach using Generative AI, specifically an enhanced ChatGPT, to provide effective PBL feedback. For an undergraduate Web Technology course, we integrated three methods: 1) fine-tuning ChatGPT with feedback from various sources; 2) using additional course-specific information for context; 3) incorporating external services for specialized feedback. We developed a tool that implements these methods both independently and in a combined fashion. We assessed the effectiveness of the tool we developed by conducting user studies, which confirmed that this tool improves the quality of feedback as compared with general-purpose ChatGPT. By acquiring and retaining knowledge from different sources, our approach offers a powerful component for implementing PBL on a large scale.

1 Introduction

Project-based learning (PBL) has been regarded as a transformative method in education. This method bridges the gap between theoretical learning and practical application¹, allowing students to engage directly with real-world problems. In PBL, students apply the concepts and skills acquired in the classroom to solve real-world problems, thereby enhancing their understanding and retention of knowledge. To make the PBL process more effective, the integration of continuous feedback at various stages of the project plays a crucial role². This feedback provided to students acts as a guiding tool, helping them to refine their approaches and encourage progressive thinking³, alongside motivating and ensuring they are moving in the right direction⁴. For courses that require both theoretical understanding and practical hands-on implementations, PBL has proven to improve students' learning outcomes⁵. For example, through PBL with

continuous integration of feedback in Computer Science courses, students learn not only how to code or design systems but also develop the necessary problem analysis and critical thinking skills to model real-world applications.⁶. Such hands-on experience helps them develop the skill sets required in this rapidly evolving field of technology.

Despite these advantages, many instructors find it challenging to fully implement PBL in their courses⁷. One major reasons is the significant amount of time and specialized knowledge needed to provide effective feedback. Additionally, the diversity in student approaches to accomplishing the same tasks demands more involvement from domain experts in the feedback process. These challenges tend to grow proportionally with an increase in student enrollment in a course, making PBL difficult to scale for larger class sizes^{8,9}.

Recent advancements in generative AI models, such as ChatGPT showcase their applications in engineering education^{10,11,12}. Such generative AI models could be applied to assist in timely feedback provisioning, reducing the reliance on domain experts and consequently enhancing the scalability of PBL. However, a recent study¹³ reveals that although students expect AI to provide feedback on their projects, they find "AI's capability quite limited, such as its ability to offer only coarse-grained analyses and its inability to tailor solutions to a specific project context."

This paper introduces our exploration in enhancing the general-purpose generative AI for efficient feedback provisioning for an undergraduate-level Web Technologies PBL course. In particular, we adapt three different methods: 1) Retaining previous knowledge of domain experts and educators - by using their feedback to fine-tune a ChatGPT model; 2) Providing relevant context from course-specific materials and previous submissions of a continuous project - by feeding them to ChatGPT using Retrieval-Augmented Generation (RAG); 3) Obtaining additional task-specific information from external data sources and services - by customizing tool functions using RAG. Combining the aforementioned methods, we developed a tool on top of the general-purpose ChatGPT that is better specialized for our particular PBL use case. Our evaluation of the tool's effectiveness involved comprehensive user studies through surveys, which confirmed that this specialized tool significantly enhances the quality of feedback compared to the general-purpose ChatGPT model. This improvement is perfectly aligned with the unique objectives and requirements of specific courses, guaranteeing that the feedback provided is not only informative but also precisely tailored to the course content. The ability of our tool to integrate and apply knowledge from diverse sources has proven to be a vital asset in facilitating PBL on a larger scale. As per our knowledge, we are the first to enhance the general-purpose ChatGPT using various methods and study their effectiveness for a PBL use case.

By describing the design, implementation, and evaluation of the proposed methods, this paper makes the following contributions:

- We introduce our methods for enhancing general-purpose generative AI and demonstrate its adaptability for integration into PBL courses by using Web Technologies as an example.
- We develop a tool that implements the proposed methods for feedback enhancement and a framework for their combinations.
- Our evaluation reveals that the feedback provided by our tool is highly rated by both the students and the instructors/ TAs, thus showcasing its effectiveness in feedback generation.

The rest of this paper is organized as follows: Section 2 briefs the background and related works on PBL; Section 3 discusses our methodology for enhancing a general-purpose generative AI model; Sections 4 and 5 detail the implementation and evaluation of our tool; Section 6 discusses the current limitations and future directions, and Section 7 concludes this paper.

2 Related Work

Feedback is a powerful tool for enhancing students' learning and achievements. It serves as a crucial bridge between existing knowledge and new learning, regardless of whether the emphasis is on guidance or on learners actively constructing knowledge. Feedback can be divided into four levels - Task, Process, Self-regulatory, and Self¹⁴. Out of all these four levels, task-level and process-level are mostly used⁹. Task-level feedback focuses on the quality and effectiveness of task completion. In contrast, process-level feedback delves into the methods and techniques used during tasks and how tasks are connected or expanded upon, such as pinpointing strategies to identify mistakes and actively learning from them⁹. Several studies in the past decade have explored strategies for providing feedback in an academic environment. These investigations have yielded a range of evidence-based suggestions to utilize feedback effectively and enhance students' motivation, achievement, and confidence. Recent research indicates that for feedback to be most effective, it needs to be: "Specific and Actionable", and "Timely and Iterative"¹⁵.

Studies have shown that project-based learning with feedback significantly enhances the effectiveness of students' learning processes¹⁶. A study was conducted on students using two distinct types of assessments, followed by a survey to determine the most beneficial assessment method¹⁷. These assessments were: 1)Summative, i.e., conducted at the end of a course or semester, and 2)Formative, i.e., carried out throughout the learning process. Their work focused on the formative assessments for project-based learning (PBL), where they integrated various feedback mechanisms, including feedback from instructors, teaching assistants (TAs), and peers, to enhance the learning process. Their evaluation through surveys revealed that 82% of students strongly agreed on formative assessments, and 79% felt that the feedback they received from these assessments significantly improved their learning. Another study¹⁸ conducted for an undergraduate-level Software Engineering course combined project-based learning with various feedback methods, including feedback from instructors, outside experts, peer students, project tools and artifacts, and self-assessment. The findings confirmed that incorporating feedback within project-based learning effectively motivated students to complete their projects within the expected timeframe.

ChatGPT is becoming a crucial tool in education, offering a new way to enhance learning. It assists students in improving their writing skills by providing feedback and suggestions for improvement. Although not yet widely used in Project-Based Learning (PBL)^{19,20}, ChatGPT can be integrated into project-based learning, enhancing interactive and personalized learning experiences. Additionally, it can assist in formative assessment by summarizing arguments and concepts, allowing educators to focus on core ideas and critical feedback²¹. The limited research on ChatGPT's role in this area highlights a need for inventive strategies that leverage its capabilities for personalized engagement in an educational environment¹³. Our approach focuses on leveraging tailored AI feedback, providing a solution to the challenge of delivering personalized, constructive, and encouraging feedback in PBL environments, thereby enhancing

the learning experience and fostering deeper engagement with the subject requirements.

3 Methodology

This section details our proposed methods for enhancing the feedback generated by a generative AI model for PBL courses. In particular, we discuss three different approaches on top of a general-purpose ChatGPT model: 1) Fine-tuning (FT), 2) Using Additional Course Relevant Context (AC), and 3) Incorporating External Services (ET). To better understand the need for each of these methods and their contributions to feedback enhancement, we applied them to a project-based learning (PBL) Web Technologies course, and employed it as a motivating example in the following subsections. Although we used Web Technologies as our motivating use case, all these methods are generic and can be integrated for feedback provisioning in other PBL courses.

3.1 Course Project Design

The Web Technologies course equips students with the necessary skills for developing interactive websites and web applications. It covers the theoretical aspects of user interface (UI) design principles and database design for data storage, as well as the use of programming frameworks for implementing business logic, etc., Alongside lectures on theory and principles, the course emphasizes practical implementations similar to other CS courses. Through PBL, students can achieve improved learning outcomes by applying theoretical knowledge in practical scenarios.

Designing this course as a PBL with the integration of continuous feedback, effectively connects theoretical concepts with practical implementation, thereby enhancing the course learning objectives . For their project, students are tasked with creating an interactive multi-user website that serves the needs of university students and faculty. Each week, the course includes 3 hours of theoretical instruction, followed by practical assignments related to these concepts, designed to equip students with the skills needed for their project work. Students enrolled in the course, typically sophomores or higher, are required to complete prerequisites in programming and software development, ensuring they have essential knowledge for the course, regardless of their university year.

To better enhance the students' learning throughout the course, we divided the entire PBL module into three checkpoints. We carefully design these checkpoints based on our initial background survey, which assessed students' familiarity with Web Technologies. This ensures that students have both the basic knowledge and sufficient time to work on them, with feedback integrated halfway through each checkpoint. The checkpoints are summarized in Table 1 and are described as below:

Checkpoint 1: Project Proposal: This checkpoint requires students to brainstorm through their project ideas, identify the main modules, better understand the functionality and operations involved with each module, design some mock-ups of the main web pages and develop a plan of their overall project.

Checkpoint 2: Front-end Design: This checkpoint requires students to design at least 3 main

Checkpoint	Purpose	Submission	Expectation	Technical Skills
Checkpoint 1	Project Proposal	Report (project idea, competitive analysis, plan) and mock-up designs	Project Idea Finalization	UI design tools
Checkpoint 2	Front-end Design	Codebase (HTML, CSS, JS, assets) and a report describing design choices	Visually appealing pages, Consistent design/ layout, Easily locatable and interactive elements	HTML, CSS, JS, Bootstrap
Checkpoint 3	Back-end Implementation	MySQL database files, Back-end codebase (PHP), and a short report documenting this	Database and Table Designs, Business Logic Implementation, Testing of the main modules	MySQL, PHP, Server Deployment

Table	1:	Checkpoint	Design
		rr	

web pages of their proposed web application using front-end technologies such as HTML, CSS, JS, and Bootstrap.

Checkpoint3: Back-end Implementation: In this checkpoint, students are expected to finalize the back-end development and rigorously test their web application. This includes designing a clear and efficient database schema, succinctly implementing the back-end business logic using appropriate scripting languages, and ensuring all aspects of the website function smoothly through comprehensive testing.

For the projects, we ask the students to work in groups of 2-4 people and provide continuous feedback to each group at different stages of the project. Halfway through each checkpoint, each team presents their work progress and further plans to the class, where they receive real-time feedback from both peer students and instructors/ TAs. Additionally, we encourage them to use ChatGPT to receive feedback on their work and provide useful prompts for doing so. Throughout the paper, we use the following input and output structure when interacting with a generative AI model: **Input**: It should consist of the prompt along with the student's submission for which feedback is being sought; **Output**: The output will be feedback on the student's submission, tailored to the provided prompt.

After the completion of Checkpoint 3, students are required to give a 5-minute presentation on their project to the class and external evaluators from the industry. We asked students to consider their final presentation as a roadshow pitch where their aim is to persuade the evaluators to invest in their project. At the end, students also received valuable feedback from the external evaluators on their overall project.

Therefore, each stage in the PBL process is meticulously designed to introduce the real-world Software Development Lifecycle (SDLC), guiding students through every phase of project development. This approach not only imparts theoretical knowledge of Web Technologies but also offers practical experience in employing these technologies to create a comprehensive web application, providing insight into the various stages of a web development project.

3.2 Fine-tuning ChatGPT

Fine-tuning is a technique that helps to make a general-purpose generative AI model (ChatGPT) better suited for a specific task with users' provided data, thus adding personalization capability to it^{22,23}. As the name suggests, this process fine-tunes the model's capabilities, such as its structure, personality, and style, to align with specific tasks.

After the fine-tuning process, the model's knowledge remains static and only changes when undergoing another round of fine-tuning with a new set of data. In our approach, we aim to make the model's feedback more tailored to the user's input, rather than providing generalized responses. For instance, consider a common webpage element, like a login form, which is frequently found in many applications. If there is a password input field within this login page, and its type is specified as "text", we want our model to recognize and address these specific details, rather than offering a generic feedback.

To achieve this goal, we explored various sources of feedback and incorporated them into our project-based Web Technology course. The explored sources include:



Figure 1: Steps involved in retaining valuable feedback

- 1. **Feedback from Instructors/TAs**: Instructor and TAs with relevant industry experience in Web Technologies were chosen to give feedback on students submissions at different stages for each checkpoint, continuing until the final submissions. They followed specific rubrics and emphasized the implementation of fundamental best practices.
- 2. **Peer Review Feedback**: In addition to TA feedback, we encouraged students from different project groups to review each other's work. To facilitate open and honest feedback, these reviews were made anonymous using Canvas's anonymous comment feature. This was implemented at each checkpoint submission presentation, focusing on high-level, perspective-driven feedback.
- 3. **Domain experts feedback**: To further enrich the evaluation process, we engaged three external domain experts as evaluators for the student's final presentations. These experts were chosen for their deep knowledge and experience in relevant fields: two of them have worked in the web development domain for over 10 years and one for over 5 years. They were tasked with 1) providing feedback on students' final presentations; and 2) grading the presentations based on key aspects such as novelty and usefulness, UI design, technical soundness, and presentation quality including the Q&A session. Notably, the external evaluators were unaware of the feedback exploration approach used.

4. **ChatGPT Feedback**: For some of the checkpoints, we provided students with the pre-designed prompts. They were encouraged to use these prompts, to seek feedback from ChatGPT on their projects. These prompts were tailored to the specific requirements of each checkpoint, which enabled ChatGPT to offer both comprehensive and detailed feedback.

Fig. 1 presents the detailed steps involved in the process of retaining valuable knowledge obtained from various sources throughout the course duration to build a more specialized generative feedback model for PBL use cases. The process starts with students first assessing the feedback they received from various sources at different stages of their project and providing a rating (out of 5) based on its effectiveness in enhancing their learning experience. The feedback ratings from the students are then used to identify the most helpful feedback. Believing that feedback highly rated by students positively impacts their learning, those are retained for future usage. This is done by constructing a dataset comprised of such highly rated feedback and using them as input in the fine-tuning process, which results in a fine-tuned model that is specialized at generating highly effective feedback for the particular PBL course.



Figure 2: Fine-Tuned Method Workflow

Once the fine-tuned model is created, it can directly be used as a standalone model in place of the general-purpose generative AI model. Fig. 2 demonstrates the workflow when using a fine-tuned model in the feedback provisioning process for PBL, where the input contains a prompt and students' submissions. Utilizing its tailored feedback data retained from previously highly rated ones, the fine-tuned model subsequently generates feedback that is specialized and targeted for a particular PBL course. The performance of this fine-tuned model depends on the number and quality of data instances used for fine-tuning. Thus, with an increase in the number of times a PBL course is conducted, it retains more valuable feedback, further improving its ability to generate specialized and effective feedback.

3.3 Providing Relevant Information related to Students' Submission

Although the fine-tuned model is effective in providing specialized feedback, when questions are asked regarding the course rubrics or previous student submissions, it cannot respond accurately, as its knowledge is confined to the data available up to its training period and does not include

specific, up-to-date information related to the course such as rubrics and students' previous checkpoint submissions.

To ensure that feedback is more closely aligned with course objectives, we incorporate the Retrieval Augmented Generation (RAG) method²⁴ into our design, providing essential information about course rubrics and previous student submissions. RAG is a framework in generative AI that enhances the response generation of a general-purpose generative AI model. It does this by integrating real-time data retrieval capability from dynamic, external sources^{24,25}.



Figure 3: Additional Context Work Flow

Fig. 3 shows the workflow of this method. The relevant information is cut into smaller pieces and converted into word vectors by applying embedding²⁶, thereby forming an external knowledge database. When a user asks a question, RAG initially analyzes it and searches the external databases for the most relevant information piece. Once it finds pertinent data, it merges it with the original user query, creating an enriched prompt. The general-purpose generative AI model then processes this enhanced query to formulate a response. This method significantly improves the original user prompt with current, relevant context, leading to a more comprehensive response.

3.4 Incorporating External Services

Although providing relevant information related to the course and students' past checkpoint submissions helps the general-purpose generative AI model to offer high-level feedback that falls within the specific requirements and limitations of the course, it still lacks the necessary context for providing specific task-level feedback. For instance, the rubrics for Checkpoint 2 in our Web Technology PBL course include a requirement for code validation, which, although is considered by the previous method, but it still lacks sufficient capability to perform code analysis on its own,

thus providing a generic response for code validation. This shortfall arises because these generative AI models do not possess a comprehensive understanding of current coding standards and practices that are essential for quality coding.



Figure 4: External Services Workflow

By integrating external services to handle such specific task-level functionality and using their results as context, we further enhance the ability of the general-purpose generative AI model to deliver more accurate and specific detailed feedback. External services here refer to any third-party services that can readily be integrated. In the case of the code validation example discussed earlier, we can use external tools to run a detailed code analysis on the students' submitted codebase and use the response obtained from the tool as context to the general-purpose generative AI model, thus providing it with extensibility capabilities and hence better enhancing the feedback.

The workflow of this method is presented in Fig. 4. When the GPT-4 agent receives a prompt and user submissions, it first uses RAG to assess and select the most relevant source that aligns with the task described in the prompt. Subsequently, the task function corresponding to the selected source is invoked, which then requests and receives a response back from this source. Finally, RAG retrieves this response and provides it as additional context to the general-purpose generative AI model for feedback generation. In this design, for each external source, there exists a corresponding tool function that incorporates necessary information from the selected source into the prompt, thereby enriching it before it is processed by the general-purpose generative AI model to generate a detailed and contextually rich completion.

4 Implementation

This section presents the implementation of the proposed methods outlined in Section 3. We developed a tool that independently implements each of these methods and later combines them to showcase its improved feedback generation capabilities. In the following subsections, for each of the three methods, we provide their respective implementation details for the tool development, along with an example observation showcasing their feedback generation capabilities. At the end, we present a combined framework where all of these methods are integrated together and developed into a single tool, alongside reporting some example observations.

4.1 Fine-tuning ChatGPT



Figure 5: Feedback generated by a fine-tuned model on HTML codebase

We developed a fine-tuned model on top of the general-purpose ChatGPT (GPCG) using the GPT-3.5 version for our Web Technologies PBL course. For this, we constructed a dataset containing valuable feedback retained from two semesters (Fall 2022 and Fall 2023) of running the Web Technologies as a PBL course. As we only considered the TAs/ instructors and domain experts feedback previously in Fall 2022, we asked the respective students from that semester to provide ratings (out of 5) on those feedback they received for each checkpoint submissions and at the end of the project. However, in Fall 2023, we integrated feedback from all four different sources (TAs/ instructors, peers, domain experts, ChatGPT) as shown in Fig. 1, and asked the students to provide ratings on those feedback.

We then combined the feedback that was highly rated by students from both of these semesters and created a dataset resulting in a total of 50 instances. This dataset is composed of pairs of prompts and completions, for instance, with the prompts being instances of users submitting their HTML code for web pages, and the completions being the insightful feedback on those submissions. Later, we used this dataset to fine-tune a GPT-3.5-turbo model²⁷.

We tested the fine-tuned model against different project submissions and found that the outcomes from the fine-tuned model tend to be more precise and directly related to the students' questions, rather than providing broad, high-level feedback. One such example response generated by our fine-tuned model is illustrated in Fig. 5, which was obtained when asking for feedback on the HTML code for a login page for one of the project teams. The highlighted sections show that the fine-tuned model was able to provide feedback on aspects such as page loading times, input types, and cursor hover behaviors, which are the feedback previously given by domain experts and instructors for other projects with similar issues.

4.2 Providing Relevant Information related to Students' Submission

6 12:38:31 PM	Can you please review and provide feedback on my project proposal annotated-GT-S Took 1 step v
12:18:09 PM	Your project proposal for the GT-System is quite comprehensive and demonstrates a clear understanding of the problem space and the needs of the Ann Arbor District Library. Here are some constructive feedback points to consider: 1. Project Overview: You've done an excellent job identifying the current challenges with the library's
	system and proposing a solution that addresses these pain points. The focus on creating a user-friendly website that integrates with existing Drupal CMS is strategic and thoughtful.
	2. Competitive Analysis: The analysis of existing products like Battlefy, Start.gg, and Challonge is detailed and well-reasoned. It's good to see you've identified why these solutions are not a perfect fit for the library's needs. However, consider also mentioning any potential challenges or limitations your proposed system might have compared to these established platforms.
	3. Unique Features: The list of unique features for the GT-System is impressive. It's great that you're considering user experience by allowing event entry without an account and focusing on library-specific taxonomy. Make sure to prioritize these features based on the most critical user needs and the project's scope.
	4. Implementation Plan: The modular approach to the project is smart, and the prioritization of modules seems logical. However, ensure that each module has a clear definition of "done" so that the scope remains manageable. It may also be beneficial to include some form of timeline or milestones to track progress against.
	 User Permissions: The user permission levels are well thought out, showing that you've considered various user roles and their access needs. Ensure that the permissions are implemented securely to protect user data and the integrity of the system.
	6. Division of Labor: The division of labor among team members is clear, and it's good to see that roles are aligned with individual strengths. Encourage the team to maintain open communication and be flexible, as roles may need to adapt as the project progresses.
	7. Agile Workflow: Adopting the Kanban Agile workflow is a good choice for managing tasks and workflow. Just ensure that the team is familiar with this methodology and that you have the tools and processes in place to support it.
	8. Feedback: It's excellent that you've sought feedback and incorporated it into your plan. Continue to seek diverse perspectives and be open to adapting your approach based on constructive criticism.
	In summary, your proposal is well-structured and shows a lot of promise. Keep a focus on user needs, maintain clear communication within your team, and be prepared to adapt as the project develops. Good luck with your project, and I'm confident that your team will do an excellent job!

Figure 6: Feedback on project proposal using relevant context

To provide course-relevant materials as additional context, our tool implements Retrieval Augmented Generation (RAG) using the LangChain framework²⁸. An essential feature of

LangChain is its chains, which adeptly merge a generative large language model (LLM) with a prompt, incorporating these elements into the framework. Additionally, we employed ChromaDB as our vector database to store course project relevant information in an embedded format. When needed, such information can be accessed using retrieval chains, which is then conveyed to the general-purpose ChatGPT as additional context, ultimately generating a feedback that is well within the scope of the course.

Our observations, as shown in Fig. 6, demonstrate that our tool implementing this method not only generates feedback regarding the submissions, but it does so by taking the checkpoint rubrics as additional context for providing a specialized feedback. To validate that our tool does have access to such relevant course materials, we asked a follow-up question to check if the submission satisfies the rubrics. In the assignment, we explicitly asked the students to compare their proposed projects with competitors, plan for implementation, and divide the workload within their team. The response in Fig. 7 shows that the feedback generated by our tool mentions about these specific rubrics points (competitive analysis, implementation plan, workload division, etc.), thus verifying that it can access such course-relevant information while generating feedback.



Figure 7: Course-relevant Additional Context (rubrics)

4.3 Incorporating External Services

As part of our implementation, we created a tool that incorporates a suite of external services, particularly for codebase analysis and design assessment tasks. The integrated code analysis tools include the W3C HTML Code Validator ¹, SQL validator ², and PHP code validator and fixer ³. These external validators perform an analysis on the codebase and provide a response with the validation result and further insights in case of failures. To assess the user interface (UI) layouts and screens, our tool integrates GPT-4 vision agent, which can assess the color theory and suggest the changes required to make the screens or layouts look more visually appealing to the users. This integration not only complements the feedback provided but also equips students with a more holistic understanding of coding standards that are critical in real-world applications.

We tested our tool against a HTML code validation task, and the response obtained is showcased in Fig. 8, where it is able to perform an HTML validation check using the HTML W3C validator

¹https://validator.w3.org/

²http://sqllint.com/

³https://github.com/PHP-CS-Fixer/PHP-CS-Fixer/



(a) W3C Validator

(b) Response after incorporating W3C Validator

Figure 8: Integrating W3C Validator as External Service

(Fig. 8a) and utilize its result as additional context for feedback generation. The highlighted section demonstrates that our tool obtained the same response for the validation check as the W3C validator, showcasing the added capability and extensibility that our tool offers to the general-purpose ChatGPT for enhancing specific task-level feedback.

For our use case, we primarily integrated the tools for code validation and design assessment. However, this approach is flexible and can be easily adapted to integrate with any other external sources by simply creating a new tool function. An interesting example could be integrating external APIs for obtaining real-time data, which further enhances the static knowledge base of general-purpose generative AI model to have a dynamic and near real-time updated context whenever required.

4.4 A Combined Framework

All three methods described earlier in Section 3 offer their distinct advantages towards enhancing a general-purpose generative AI model. Given this, our objective is to integrate them together to further explore the feedback improvement that can be achieved using such a collaborative approach. By combining the strengths of each method, we aim to create a more comprehensive and effective feedback generation framework on top of the general-purpose generative AI model.

In this framework, we integrate the previous methods (Fine-tuning, Relevant Course Context, and External Services) to assess their collective impact on feedback enhancement. This comprehensive approach aims to boost the general-purpose generative AI model's ability to deliver feedback that is not only precise and highly relevant to the course objectives but also is specialized in the level of specific tasks.

Fine-tuning + Additional Context + External Tools

The components and workflow of this framework are presented in Fig. 9. When a student submits a prompt along with their submissions, it is first routed to RAG, which acts as an entry point to



Figure 9: Combined Framework

the system. RAG, upon analyzing the prompt, determines which of the three methods is required to provide feedback. Based on the decision, RAG activates the necessary approaches by issuing a request for further context. Each of the activated methods undergoes its own workflow, as described earlier in Section 3, and sends the respective context back to RAG, which is then passed to the generative model for feedback generation.

This combinatorial approach provides a one-stop mechanism for feedback enhancement and personalization. As observed in Fig. 10, when users request the model to simultaneously handle three distinct tasks – providing code feedback, checking rubrics, and performing code validation – this approach is designed such that it can consider all those distinct contexts and provide fine-grained feedback covering all the aspects. For such multifaceted inputs, we have incorporated fine-tuning specifically for fine-grained code feedback, provided rubrics information as course-relevant context, and utilized external services to obtain further task-level details for feedback generation.

It is not compulsory to integrate all approaches every time. For example, when a user seeks feedback on a project proposal, invoking external services and fine-tuning models would be unnecessary; simply providing course-relevant context is adequate to yield a valuable feedback. This is because the fine-tuned model is better specialized to provide lower-level technical feedback and its knowledge base may be limited to only previously submitted proposal ideas, whereas, the general-purpose model, despite its ability to cover new ideas with its large knowledge base, may go beyond the scope of the course. In this scenario, just integrating the general-purpose model with the course rubrics allows the model to explore new ideas while being restricted within the course's scope, offering a perfect match.

Such flexibility of our combined framework allows for tailored responses by applying a combination of the most appropriate methods depending on the input request, thus ensuring that the feedback generated is both relevant and efficiently tailored to the course project requirements.



Figure 10: Feedback using (Fine-tuning + Additional Context + External Services)

5 Evaluation

We evaluate the effectiveness of our proposed solution by examining how its generated feedback aligns with and supports course objectives and requirements. For this purpose, we conducted a user survey and quantitatively analyzed the results.

5.1 Questionnaire Design and Participants

The survey was mainly designed to provide a quantitative analysis on various feedback methods by evaluating their effectiveness for feedback generation. It included 7 sets of unique questions, each presenting a prompt and student's submission as input, and the obtained feedback as output. Each of these questions was tested against different methods, as mentioned in Section 3, resulting in a total of 18 different questions. Table 2 presents the set of questions and the methods used for obtaining a response. We randomly selected three different student project submissions from various checkpoints and supplied the same set of 18 questions to each of them. The responses were recorded and sent out to the group members of the respective projects, as well as the TAs and instructor. Both the students and TAs/ instructors were asked to provide a rating on a scale of 1 to 5 on the collected feedback for each of the questions. To remove bias, we ensured that the participants were unaware of the different methods used in generating the feedback. Overall, we collected answers from 11 participants, with 2 being instructors and 9 being students.

	Questions	Comparision of methods
Q1	Asking for feedback on project proposal	1. General-Purpose ChatGPT (GPCG)
Q2	Asking if Q1 meets the rubrics require-	2. Additional Context (AC)
	ments as a follow-up question.	
Q3	Asking for feedback on codebase	1. General-Purpose ChatGPT (GPCG)
Q4	Asking if Q3 meets the rubrics require-	2. Fine-tuning (FT)
	ments as a follow-up question.	3. Additional Context (AC)
Q5	Asking for code validation and feedback	1. General-Purpose ChatGPT (GPCG)
Q6	Asking for feedback on visual elements and	2. Fine-tuning (FT)
	UI designs	3. External Services as Tools (ET)
		1. Fine-tuning (FT) + Additional Context
Q7	Asking for feedback based on rubrics re- quirements and code validation	(AC)
		2. Fine-tuning (FT) + Additional Context
		(AC) + External Services as Tools (ET)

Table 2: Survey Questionnaire

5.2 Effectiveness of various Feedback Methods for PBL

The survey responses are represented through histograms, which illustrate the distribution of ratings for various feedback methods. These histograms serve as a tool to uncover patterns and trends from the collected data, showing whether a significant number of users preferred a certain method or if the ratings were more evenly distributed. This analysis helps in assessing the preferences and effectiveness of each feedback method. Moreover, examining the histograms' shapes allows us to see the direction of skewness in the responses, indicating whether there was a general tendency for higher or lower ratings. This aspect is vital for understanding the overall effectiveness and acceptance of the feedback methods among users. The skewness value provided for each histogram quantifies the distribution's asymmetry, offering a precise measure of how users ratings deviate from a balanced distribution. Observations regarding the performance of various methods across different tasks are provided below.

Assessing Feedback: GPCG vs AC methods for Project Proposal

The feedback ratings for questions Q1 and Q2, when analyzed for the GPCG and AC methods as shown in the histograms 11a, 11b, 11c, 11d, exhibit distinct trends. Both Q1-GPCG and Q1-AC have approximately symmetrical feedback distributions with skewness values of 0.11, though Q1-AC has a slight bias towards higher ratings, and Q1-GPCG shows a neutral pattern. In contrast, Q2-AC shows a skewness of -0.96, reflecting a strong preference for higher ratings, while Q2-GPCG, with a skewness of 0.96, tends towards lower ratings.

Observation 1

The Additional Context(AC) method, providing course-relevant information, is better for generating feedback on project proposals as it not only offers feedback but also responds to questions regarding proposal rubrics related to the course, which cannot be answered by ChatGPT(GPCG).





Figure 11: Effectiveness of various feedback methods (Q1 - Q7)

Assessing Feedback: GPCG vs FT vs AC methods for Codebase

For the questions Q3 and Q4, feedback ratings are evaluated using three distinct methods: GPCG, FT and AC. The graphs 11e,11f,11g,11h,11i and 11j indicate a general trend of positive response. Q3-AC and Q4-GPCG, with skewness of 0.13 and 0.18, displayed a balanced distribution, and Q3-FT and Q4-AC, with skewness values of -0.80 and -1.57, showed a clear tendency towards higher ratings.

Observation 2

FT demonstrates greater performance over GPCG and AC in offering code feedback, as it has been trained on a customized dataset. This specialized training enables the FT model to deliver more insightful feedback on coding tasks.

Assessing Feedback: GPCG vs FT vs ET methods for Code Analysis

The evaluation of Question Q5, using methods GPCG, FT, and ET, is depicted in the graphs 11k,11l, and 11m respectively, revealing distinct skewness patterns. Q5-ET with skewness of -0.49 shows a tendency towards higher ratings, indicating a positive response. In contrast, Q5-GPCG with a skewness of 0.25 suggested a modest preference for lower ratings. Meanwhile, Q5-FT, with a skewness of -0.02, displayed an even distribution of ratings.

Observation 3

The ET method surpasses both GPCG and FT in terms of code validation effectiveness. This is because GPCG's capabilities are constrained to its initial generic knowledge base, while FT lacks previously retained knowledge specific to code validation.

Assessing Feedback: GPCG vs FT vs ET methods for Visual Elements and UI Designs From the analysis of histograms 11n,11o, and 11p for question Q6, each reflected the respective skewness values of -0.18,-0.11, and -0.11, indicating positive feedback with a modest lean towards higher ratings. The Q6-GPCG shows slightly stronger higher ratings than Q6-FT and Q6-ET, which are almost identical in distribution.

Observation 4

GPCG works better for giving feedback on visual elements and UI designs when compared to ET(GPT4 vision agent) and FT. Unlike the GPT-4 vision agent, which is specialized for image recognition and analysis, and the FT models that are tailored to specific datasets, GPCG offers a broad understanding of design aesthetics and user experience. This allows it to deliver more insightful feedback on visual designs.

Assessing Feedback: Combined Methods (FT+AC and FT+AC+ET)

The graphs 11q and 11r illustrate that various methods were integrated for evaluating Question Q7. The analysis of the histograms indicates distinct patterns of positive feedback: the combined approach of Q7-FT+AC+ET shows a strong bias towards the highest feedback rating with a skewness of -1.14, while Q7-FT+AC, with a skewness of -0.49, also displays a tendency towards higher ratings but with a wider range of responses.

Observation 5

Combining all three methods yields better feedback results than employing each method independently. This is due to the combined approach's ability to answer any question within the scope of the project requirements and provide better, relevant feedback with its access to additional context from relevant course materials and external services.

6 Discussion and Future Works

This paper presents a novel design approach and its quantitative investigation into the potential future of utilizing generative AI for assistance in feedback provisioning for PBL courses. Our work involves the development of a tool on top of the general-purpose ChatGPT for a PBL use case in an undergraduate Web Technologies course. Our evaluation was based on assessing the quality of feedback generated by our tool on various checkpoint submissions for three different students' project, where a total of 11 participants, including both students and TAs/instructors, provided ratings on the generated feedback. Given the limited number of participants, it is difficult to generalize our tool's effectiveness for a large class size with varying project ideas and implementations. Furthermore, the limited knowledge and experience of the participants with using a generative AI model might lead to participants not asking the right questions, impacting the quality of the feedback.

To the best of our knowledge, this paper is the first to showcase the possibilities of enhancing a generative AI model for specific PBL use cases in feedback provisioning, and motivates future research. Based on our observations, one possible future working direction could include designing a new PBL course curriculum that not only integrates generative AI for feedback provisioning but also provides student with sufficient background on it so that they consider it as a learning companion rather than completely relying on it, thus excelling the overall learning process. To further investigate the effectiveness of this approach on PBL's scalability, we plan to conduct a large-scale study using larger student and project samples, alongside comparing the domain experts effort, as part of our future work. Furthermore, we believe that our novel methodology presented in this paper has the potential to be developed into a dynamic and adaptable learning framework and envision its direct integration with learning management systems (i.e., Canvas) for automated feedback provisioning in the long run. This integration is pivotal as it facilitates the immediate provision of feedback to students on their projects with less to no need for domain experts' effort, thus boosting PBL's scalability.

7 Conclusion

The paper presents an exploration aimed at improving the general-purpose ChatGPT model to provide effective feedback for a Project-Based Learning (PBL) course. Three distinct methods were employed independently and later combined to develop a specialized tool on top of the general-purpose ChatGPT, enhancing feedback quality for an undergraduate-level Web Technologies PBL course. A quantitative analysis based on user surveys confirmed the tool's effectiveness, showing significant improvement in the generated feedback over the general-purpose ChatGPT model. The tool's ability to integrate and retain knowledge from diverse sources proves crucial for facilitating PBL on a larger scale, making this work a pioneering effort in enhancing general-purpose ChatGPT for PBL in Computer Science education.

Acknowledgment

This research is supported by NSF through grant #2104337 and UMDearborn CECS PBL/EML Grant.

References

- [1] Lu Zhang and Yan Ma. A study of the impact of project-based learning on student learning effects: a meta-analysis study. *Frontiers in Psychology*, 14, 07 2023. doi: 10.3389/fpsyg.2023.1202728.
- [2] Wilhelm A Friess and Andrew J Goupee. Using continuous peer evaluation in team-based engineering capstone projects: A case study. *IEEE Transactions on Education*, 63(2):82–87, 2020.
- [3] Rhonda Dookwah and Gabriel Julien. Motivation heightens and independent thinking deepens: Undergraduate

students share their experiences of pbl while learning microeconomics. *Educational Research and Reviews*, 17 (8):227–233, 2022.

- [4] Kathy Smith, Nicoleta Maynard, Amanda Berry, Tanya Stephenson, Tabetha Spiteri, Deborah Corrigan, Jennifer Mansfield, Peter Ellerton, and Timothy Smith. Principles of problem-based learning (pbl) in stem education: Using expert wisdom and research to frame educational practice. *Education Sciences*, 12(10):728, 2022.
- [5] Dimitra Kokotsaki, Victoria Menzies, and Andy Wiggins. Project-based learning: A review of the literature. *Improving Schools*, 19(3):267–277, 2016. doi: 10.1177/1365480216659733.
- [6] Simone C dos Santos, Priscila BS Reis, Jacinto FS Reis, and Fabio Tavares. Two decades of pbl in teaching computing: a systematic mapping study. *IEEE transactions on education*, 64(3):233–244, 2020.
- [7] Kristian Aga. Challenges and motivation for teachers transitioning to active learning spaces. *European Journal* of Engineering Education, pages 1–23, 2023.
- [8] João Paulo Fernandes, Ricardo Araújo, and Mário Zenha-Rela. Achieving scalability in project based learning through a low-code platform. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, pages 710–719, 2020.
- [9] Alaa Jaber, Kimberly Lechasseur, Khairul Mottakin, and Zheng Song. A study report in the web technologies course: What makes feedback effective for project-based learning? In 2023 ASEE Annual Conference & Exposition, 2023.
- [10] Junaid Qadir. Engineering education in the era of chatgpt: Promise and pitfalls of generative ai for education. In 2023 IEEE Global Engineering Education Conference (EDUCON), pages 1–9. IEEE, 2023.
- [11] Wei Dai, Jionghao Lin, Hua Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gašević, and Guanliang Chen. Can large language models provide feedback to students? a case study on chatgpt. In 2023 IEEE International Conference on Advanced Learning Technologies (ICALT), pages 323–325. IEEE, 2023.
- [12] Mohammad Abu-Ghuwaleh and Rasheed Saffaf. Integrating ai and nlp with project-based learning in stream education. 2023.
- [13] Chengbo Zheng, Kangyu Yuan, Bingcan Guo, Reza Hadi Mogavi, Zhenhui Peng, Shuai Ma, and Xiaojuan Ma. Charting the future of ai in project-based learning: A co-design exploration with students. arXiv preprint arXiv:2401.14915, 2024.
- [14] John Hattie and Helen Timperley. The power of feedback. *Review of Educational Research*, 77(1):81–112, 2007. doi: 10.3102/003465430298487. URL https://doi.org/10.3102/003465430298487.
- [15] Alison C. Koenka and Eric M. Anderman. Personalized feedback as a strategy for improving motivation and performance among middle school students. *Middle School Journal*, 50(5):15–22, 2019. doi: 10.1080/00940771.2019.1674768.
- [16] Berta de la Torre-Neches, Mariano Rubia-Avi, Jose Luis Aparicio-Herguedas, and Jairo Rodríguez-Medina. Project-based learning: an analysis of cooperation and evaluation as the axes of its dynamic. *Humanities and Social Sciences Communications*, 7(1):1–7, 2020.
- [17] Rania Al-Hammoud and Ona Egbue. The role of feedback in enhancing students' learning experience: An evaluation of student perspectives and attitudes. In *2023 ASEE Annual Conference & Exposition*, 2023.
- [18] Valentin Razmov and Stani Vlasseva. Feedback techniques for project based courses. In 2004 Annual Conference, pages 9–608, 2004.
- [19] Marta Montenegro-Rueda, José Fernández-Cerero, José María Fernández-Batanero, and Eloy López-Meneses. Impact of the implementation of chatgpt in education: A systematic review. *Computers*, 12(8), 2023. ISSN 2073-431X. doi: 10.3390/computers12080153. URL https://www.mdpi.com/2073-431X/12/8/153.

- [20] Vaughan Connolly and Steve Watson. Chatgpt and education.
- [21] Iwan Purnama, Firman Edi, Syahrul, Risnawati Agustin, and Nuridin Widya Pranoto. Gpt chat integration in project based learning in learning: A systematic literature review. *Jurnal Penelitian Pendidikan IPA*, 2023. URL https://api.semanticscholar.org/CorpusID:267231259.
- [22] Toan Nguyen-Mau, Anh-Cuong Le, Duc-Hong Pham, and Van-Nam Huynh. An information fusion based approach to context-based fine-tuning of gpt models. *Information Fusion*, 104:102202, 2024. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2023.102202. URL https://www.sciencedirect.com/science/article/pii/S1566253523005183.
- [23] Patent claim generation by fine-tuning openai gpt-2. World Patent Information, 62:101983, 2020. ISSN 0172-2190. doi: https://doi.org/10.1016/j.wpi.2020.101983. URL https://www.sciencedirect.com/science/article/pii/S0172219019300766.
- [24] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*, 2022.
- [25] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2023.
- [26] Koh Matsuda and Ian Frank. Langchain unleashed: Advancing education beyond chatgpt's limits.
- [27] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint* arXiv:2303.10420, 2023.
- [28] Oguzhan Topsakal and Tahir Cetin Akinci. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *Proceedings of the International Conference on Applied Engineering* and Natural Sciences, Konya, Turkey, pages 10–12, 2023.