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

by

Venkata Alekhya Kusam

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Computer and Information Science)
in the University of Michigan–Dearborn
2024

Master's Thesis Committee:

Assistant Professor Zheng Song, Chair
Professor Qiang Zhu
Lecturer III Khalid Kattan

Venkata Alekhya Kusam

alekhyak@umich.edu

ORCID iD: 0009-0000-8766-6769

Dedication

This work is dedicated to my professor Dr. Zheng Song. Your guidance, wisdom and unwavering belief in my abilities have been the cornerstones of my progress. Your mentorship has not only shaped my academic endeavors but also my personal growth. Your insights and encouragement have been invaluable and lighting the way through the most challenging phases of this journey.

To my friends and lab mates who have stood by me offered their support and shared in the ups and downs, this achievement is as much yours as it is mine. Your constant presence and belief in me have been sources of motivation and strength reminding me of the importance of community and collaboration. Thank you all for being a part of my journey.

Acknowledgements

I express my sincere gratitude to my advisor and mentor Dr. Zheng Song, whose steadfast support and profound insights were crucial in the completion of my master's thesis. Dr. Song's expertise, encouragement and commitment were instrumental in guiding my research path, enriching my academic journey and ensuring the successful completion of this significant milestone

I would like to extend my heartfelt gratitude to Dr. Qiang Zhu and Dr. Khalid Kattan for their generous contributions of time and expertise as members of the examination committee for my master's thesis. Their insightful feedback and valuable guidance have significantly improved the quality of my research. I deeply appreciate their dedication to academic excellence and their support throughout this process.

I extend my heartfelt gratitude to Summit Shrestha, Larnell Moore, Jai Ram Reddy Bollareddy, Krishna Sureddy, Varshitha Sannareddy, and all my lab mates for their support and encouragement throughout my thesis journey. Their invaluable collaboration, insightful discussion and constant encouragement fueled my motivation and inspired me to navigate through challenges and driving me towards academic excellence.

Preface

My journey into the world of Generative AI and Project-Based Learning (PBL) started with my Master's degree. The fascination with how AI could transform educational methodologies especially in computer science caught my attention and I found myself diving into the literature and exploring the vast potential of AI in enhancing learning experiences through real-world projects. Before this, AI was a whole new world to me. I had no background in it, but I was ready for the challenge. Diving into generative AI and large language models showed me just how much there is to learn and that tackling tough problems is part of the journey and offering lessons that go beyond just academics. I am super grateful for this journey for all the challenges, everything I have learned, and for all the amazing people who have been part of it

Venkata Alekhya Kusam - 3/31/2024

Table of Contents

Dedication . . . . . **ii**

Acknowledgements . . . . . **iii**

Preface . . . . . **iv**

List of Figures . . . . . **vii**

List of Tables . . . . . **viii**

Abstract . . . . . **ix**

Chapter

1 Introduction **1**
- 1.1 Problem Statement 1
- 1.2 Research Objective 2
- 1.3 Methodology 3
- 1.4 PBL Course Module Design for Web Technologies 4

2 Literature Overview **8**
- 2.1 Project Based Learning(PBL) 8
- 2.2 Feedback in PBL 9
- 2.3 Generative AI (GAI) in education 10

3 Activity 1: Enhancing Student Capability of Seeking Feedback from Generative AI, and evaluating the effectiveness of such feedback **12**
- 3.1 Methodology 12
  - 3.1.1 Introducing Prompt Engineering to Students 13
  - 3.1.2 Providing Example Prompts for Getting Feedback on Projects 15
- 3.2 Evaluation and Results Analysis 15

4 Customizing GAI for for Feedback Provisioning **21**
- 4.1 Methodology 22
  - 4.1.1 Fine-tuning ChatGPT 22
  - 4.1.2 Providing Relevant Information related to Students' Submission 25
  - 4.1.3 Incorporating External Services 26
- 4.2 Implementation 27
  - 4.2.1 Fine-tuning ChatGPT 28

4.2.2 Providing Relevant Information related to Students' Submission 29
4.2.3 Incorporating External Services 30
4.2.4 A Combined Framework 32
4.3 Evaluation 34
4.3.1 Questionnaire Design and Participants 34
4.3.2 Effectiveness of various Feedback Methods for PBL 35

5 Conclusion and Future Directions 40

Appendix: Relevant Paper Publications 42

References 46

List of Figures

1.1 Methodology Overview 7

3.1 Prompt Formula 14
3.2 Teachinhg Evaluation Results 17
3.3 Student Ratings 18
3.4 Student Ratings 19

4.1 Steps involved in retaining valuable feedback 23
4.2 Fine-Tuned Method Workflow 24
4.3 Additional Context Work Flow 25
4.4 External Services Workflow 27
4.5 Feedback generated by a fine-tuned model on HTML codebase 28
4.6 Feedback on project proposal using relevant context 30
4.7 Course-relevant Additional Context (rubrics) 31
4.8 Integrating W3C Validator as External Service 31
4.9 Combined Framework 32
4.10 Feedback using (Fine-tuning + Additional Context + External Services) 34
4.11 Effectiveness of various feedback methods (Q1 - Q7) 38

List of Tables

1.1 Checkpoint Design 5

3.1 Course Evaluation Summary 17

4.1 Survey Questionnaire 35

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 required for effective feedback, especially with increasing student numbers. This paper presents our explorations of how to better utilize Generative AI, such as ChatGPT, to assist in providing feedback in PBL. For an undergraduate Web Technology course, we developed two approaches: 1) developing a mini-course module to teach students how to obtain more effective feedback for their projects; and 2) customizing a tool that enhances ChatGPT with the following three strategies: 2.1) fine-tuning ChatGPT with feedback from various sources; 2.2) using additional course-specific information for context; 2.3) incorporating external services for specialized feedback. We assessed the effectiveness of these two approaches by conducting user studies and reported the assessment results. We found that 1) although students frequently use generative AI, providing them with additional knowledge about prompt engineering helps them more efficiently access useful information from ChatGPT; 2) our customized tool improves the quality of feedback compared with general-purpose ChatGPT. In conclusion, integrating generative AI into PBL can facilitate its implementation on a large scale, thus helping to eliminate inequity in education.

CHAPTER 1

Introduction

1.1 Problem Statement

Project-based learning (PBL) has been regarded as a transformative method in education. This method bridges the gap between theoretical learning and practical applications [1]. This pedagogical approach engages students in real-world challenges fostering not only a deeper understanding of the subject matter but also critical thinking, collaboration, and problem-solving skills [2]. Despite its benefits, including enhanced student motivation and improved learning outcomes [3], the implementation of PBL encounters significant obstacles which are primarily related to feedback provision.

Effective feedback is a cornerstone of PBL, guiding students through the iterative process of project development and refinement. It encourages self-assessment, deepens understanding and ensures alignment with learning objectives [4]. However, the resource-intensive nature of personalized feedback, which requires considerable time and domain-specific expertise poses a significant challenge, particularly for large-scale classes in computer science programs [5].

Recent advances in Generative AI, exemplified by models such as General-purpose ChatGPT, present a platform for addressing the feedback challenges in PBL [6]. These models have shown potential in various educational applications, from assisting in the generation of learning materials to providing preliminary feedback on student submissions [7]. However, limitations in the specificity and adaptability of feedback provided by such AI tools have been noted, particularly in terms of tailoring feedback to the unique context and requirements of specific projects [8]. This challenge lies in enhancing the capabilities of Generative AI models to provide feedback that is not only timely but also contextually relevant, personalized and aligned with the learning objectives of PBL in computer science courses.

This problem statement highlights the need for a novel approach that leverages the power of Generative AI, particularly General purpose ChatGPT, to tackle the challenges associated with providing feedback in PBL. Addressing these challenges, the study is poised to unlock the potential of PBL using Generative AI within computer science education, promoting its wider application and leading to significant improvements in learning outcomes.

1.2 Research Objective

The primary objective of this research is to enhance the effectiveness and efficiency of Project-Based Learning (PBL) in computer science education through the development of an innovative feedback provisioning tool that utilizes Generative AI technology. This tool is designed to improve feedback mechanisms and facilitate the wider adoption of Project-Based Learning (PBL), ensuring the quality of feedback and educational outcomes remains high.

To achieve this objective, the research focused on three main goals:

1. **Role of General-Purpose ChatGPT in Feedback Provisioning and Its Limitations**: General-purpose ChatGPT offers a new way to help with project-based learning by providing feedback that can handle many students at a time. However, there's a problem: it delivers generalized feedback. It doesn't offer the detailed or specific feedback that students need for their unique projects. This means the feedback might not be very helpful in guiding students step-by-step through their learning projects, which need more tailored advice and insights.

2. **Enhancing General-Purpose ChatGPT for Customized Course-Specific Feedback Provisioning in PBL**: To overcome the limitations of General-purpose ChatGPT in PBL, an enhanced ChatGPT model was developed through a comprehensive approach. This included

fine-tuning ChatGPT with domain-specific feedback, incorporating course-specific contexts, and integrating external services for specialized feedback. This enhancement process aimed to tailor the feedback mechanism to the unique requirements of specific courses and projects within the PBL framework. The customized ChatGPT model demonstrated improved capabilities in providing feedback that is not only relevant but also contextually rich, effectively bridging the gap between general-purpose AI feedback and the specific needs of PBL.

3. **Effectiveness of the Customized Generative AI Tool in Providing Feedback**: We evaluated how well the GAI-based tool worked for providing feedback in project-based learning by conducting user surveys. The results showed that this GAI-based tool is better at delivering useful feedback that aligns with the course objectives and requirements. This indicates that tailoring AI tools to fit specific educational needs can significantly help in providing customized feedback to students.

Overall, our research shows that by carefully enhancing AI tools like ChatGPT to provide more specific feedback, we can significantly improve the learning experience in a project-based learning environment. This approach not only makes feedback more relevant and helpful for students but also supports the wider use of PBL in computer science education.

1.3 Methodology

To fulfill these objectives, we designed two research activities, as introduced below:

1. **Activity 1: Enhancing Student Capability of Seeking Feedback from Generative AI, and evaluating the effectiveness of such feedback**: We developed a mini-course module focused on teaching students how to craft prompts for ChatGPT. Subsequently, we asked the students to use these prompts to obtain feedback from ChatGPT on specific assignments. Our evaluation scrutinized the utility of the feedback provided by ChatGPT. Insights from this study informed our investigation of the first research question regarding the ways ChatGPT

can assist in providing feedback and its limitations. The details of this study and its findings are presented in Chapter III.

2. **Activity 2: Customizing General-Purpose Generative AI to Meet PBL Needs**: To tailor General-Purpose Generative AI (GAI) for Problem-Based Learning (PBL), we implemented three strategies: 1) Incorporating the prior knowledge of domain experts and educators through their feedback to fine-tune a ChatGPT model; 2) Integrating relevant context from course-specific materials and previous submissions of a continuous project by inputting them into ChatGPT using Retrieval-Augmented Generation (RAG); 3) Acquiring additional task-specific information from external data sources and services by customizing tool functions using RAG. By combining these methods, we developed a tool built upon the general-purpose ChatGPT, better specialized for our specific PBL use case. The effectiveness of these approaches was evaluated through student surveys, and the insights derived from this study were utilized to address research questions 2 and 3. The details and outcomes of this study are discussed in Chapter IV.

These approaches and studies were conducted within the context of an undergraduate course on “Web Technologies“. In the following section, we provide a detailed description of the Project-Based Learning (PBL) design implemented for this course.

1.4 PBL Course Module Design for Web Technologies

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

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 1.1: Checkpoint Design

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 1.1 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 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.

The overview of methodology is shown in the figure1.1. This workflow shows how the progression of chapter III and IV

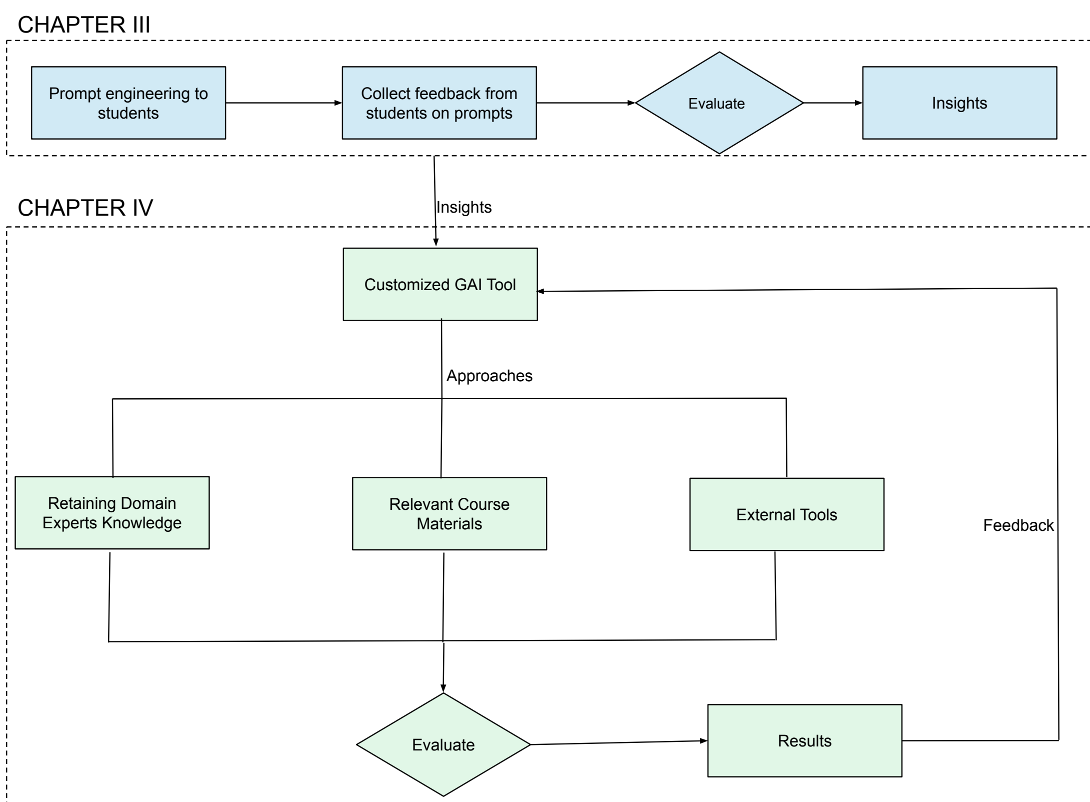


Figure 1.1: Methodology Overview

CHAPTER 2

Literature Overview

In this chapter, we introduce the related work and background for this thesis. In particular, we surveyed PBL and its implementations, the feedback problem and current solutions, as well as the application of generative AI in education, which is an emerging field.

2.1 Project Based Learning(PBL)

Project-based learning (PBL) is widely recognized as a transformative educational approach, effectively closing the gap between theoretical concepts and their practical application [1], 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 role [9]. This feedback provided to students acts as a guiding tool, helping them to refine their approaches and encourage progressive thinking [3], alongside motivating and ensuring they are moving in the right direction [10]. For courses that require both theoretical understanding and practical hands-on implementations, PBL has proven to improve students' learning outcomes[11]. 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. [12]. 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 courses [5]. 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 [13, 14].

2.2 Feedback in PBL

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 Self [15]. Out of all these four levels, task-level and process-level are mostly used [14]. 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 them [14]. 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” [16].

Studies have shown that project-based learning with feedback significantly enhances the effectiveness of students' learning processes[17]. A study was conducted on students using two distinct types of assessments, followed by a survey to determine the most beneficial assessment method[18]. 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 study[19] 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.

Overall, feedback in project-based learning is essential for guiding students through their tasks and helping them learn from the process. By focusing on specific, actionable, and timely feedback, educators can significantly enhance student engagement, achievement, and confidence.

2.3 Generative AI (GAI) in education

Incorporating advanced AI models like ChatGPT into engineering education is reshaping how we approach teaching and learning. These technologies offer new ways to enhance how students learn, especially in project-based learning (PBL) settings. Feedback is crucial in learning, especially in PBL where students solves complex issues. Traditionally, providing feedback depends on instructors time and expertise, which can slow down the process. AI models like ChatGPT promise to change that by offering instant and useful feedback, helping students navigate through their projects more smoothly. Recent advancements in generative AI models, such as ChatGPT showcase their applications in engineering education [6, 7, 20]. 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 study [8] 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. Despite these challenges, the potential of ChatGPT and similar AI tools in education cannot be understated. Their ability to provide immediate feedback can significantly enhance the learning experience, offering students insights and suggestions that might not be readily available through traditional means. In the realm of PBL, this could translate to more dynamic learning environments where students receive constant guidance and support, thereby enhancing their engagement and motivation. Moreover, the integration of AI in PBL can facilitate a more personalized learning experience [21, 22]. By analyzing student submissions and providing tailored feedback, AI tools can help identify areas of strength and weakness, enabling students to focus their efforts more effectively. This personalized approach not only benefits students by providing them with targeted feedback but also assists educators by highlighting areas where additional instruction or support may be needed [23].

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)[21, 22], 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 feedback[23]. The journey toward integrating generative AI into PBL is an ongoing process that demands continuous research, development, and collaboration. As AI technology evolves, so too will its applications in education. By harnessing the capabilities of generative AI models, educators can create more engaging, effective, and scalable PBL environments that prepare students for the challenges of the modern world, So 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 environment[8].

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.

CHAPTER 3

Activity 1: Enhancing Student Capability of Seeking Feedback from Generative AI, and evaluating the effectiveness of such feedback

Despite most students already having experience using Generative AI in their educational context [24], surveys suggest that students lack knowledge about how to customize their questions to get the most desired results from GAI. To bridge this gap and verify whether GAI, in its full potential, can provide sufficient feedback on students' PBL activities, we first teach students prompt engineering as a way to more effectively interact with GAI and evaluate its impacts and limitations.

3.1 Methodology

In our course module on Project-Based Learning (PBL) for web technology, we encouraged students to seek feedback from the general purpose ChatGPT at various checkpoints. However, we encountered an issue: the feedback from ChatGPT was often too generic and did not meet their specific needs at these checkpoints. This issue emerged because ChatGPT is trained on a vast dataset, which makes it challenging for it to provide targeted responses.

To address this challenge, we explored how students could interact more effectively with ChatGPT to elicit more relevant feedback. Through this exploration, we discovered the concept of prompt engineering. We found that by applying prompt engineering techniques when engaging with ChatGPT, the responses became more specific to the students' requirements, contrasting with the generic responses received from direct questioning.

As a result, we introduced prompt engineering to the students, teaching them how to effectively utilize it to obtain more useful feedback for their checkpoints using general purpose ChatGPT.

3.1.1 Introducing Prompt Engineering to Students

Our primary objective in introducing prompt engineering to students is to teach them the correct way to ask questions of AI systems like ChatGPT. This skill is crucial as it enables students to maximize the benefits of AI, whether they need assistance with assignments, are seeking creative ideas, or tackling complex problems. By focusing on effectively crafting their questions, students can guide the AI to provide the most accurate and useful answers. This skill is an important part of their education, preparing them for a future where AI is a ubiquitous tool across many jobs and industries. It is essential due to the nature of large language models, which are developed with vast amounts of textual data. These models can sometimes interpret queries in unexpected ways making it challenging to achieve the intended output. Thus, learning to craft and refine prompts through prompt engineering becomes crucial to ensure we obtain the desired results for our tasks.

Prompt Engineering involves the creation and refinement of prompts, which are the instructions and context provided to large language models (LLMs) to accomplish specific tasks [25]. These prompts are essential for the effective utilization of LLMs. It's not only about asking questions but also about understanding how AI language models work, learning how to ask questions that will elicit the most useful answers from LLMs, and grasping the context in which the information will be used [26]. This expertise is crucial for eliciting useful and comprehensive responses from LLMs significantly enhancing the feedback generation and learning process in PBL scenarios.

1. **What are Prompts?:** Prompts involve instructions and context passed to a large Language model to achieve a desired task [27].

2. **What is Prompt Engineering?:** Prompt Engineering is the practice of developing and optimizing prompts to efficiently use language models(LLMs) [27]

In our lecture on prompt engineering, we covered a wide range of essential topics along with example usages that provided a basic understanding for the students. We began by defining what prompt engineering is and why it's becoming increasingly crucial in today's AI-driven world. This initial discussion helped students grasp the significance of communicating effectively with AI systems like Large Language Models (LLMs), highlighting the role of prompt engineering in enhancing the efficiency and accuracy of AI responses. Later, we discussed different prompting types, roles, and patterns, along with examples for each case. Students learned how the structure and phrasing of prompts can significantly influence the output. We discussed strategies for creating effective prompts, emphasizing the importance of clarity, specificity, and context in formulating questions that lead to meaningful and useful AI interactions. After the class, we asked students to use prompt engineering techniques for their own use cases. Many students then began using these techniques while interacting with ChatGPT.

From all the prompt engineering techniques we explored in the course, we have developed a formula to guide students in eliciting desired responses from the general-purpose ChatGPT for feedback on their project checkpoints. The structure of this formula is illustrated in Figure 3.1.

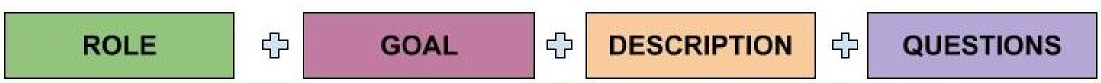


Figure 3.1: Prompt Formula

In this formula, 'Role' refers to the specific style or persona you want the AI to adopt. 'Goal' is about setting a clear objective for the AI to achieve. 'Description' involves providing all necessary context in the prompt. Lastly, 'Questions' allow the AI to seek clarification on any uncertainties before responding.

Equipped with this knowledge, students gain the skill to ask the right questions when interacting with LLMs, ensuring that they communicate with the AI in a clear and concise manner. By posing the right questions, they can guide ChatGPT to provide targeted and helpful feedback that significantly aids in their project work. Along with the prompt engineering lecture, we introduced customization of large language models using fine-tuning and Retrieval-Augmented Generation.

These concepts help them understand how they can customize large language models for their specific applications and requirements.

3.1.2 Providing Example Prompts for Getting Feedback on Projects

We provided students with specific prompts, similar to the example below, for Checkpoints 1 and 3 to obtain feedback on their submissions from a general-purpose ChatGPT model. However, for Checkpoint 2, we took a different approach. We did not supply any prompts. Instead, we encouraged students to independently use the general-purpose ChatGPT to seek feedback on their work. The example prompt provided for Checkpoint 1 is given below.

Example: Act as a web technology professor well-versed in HTML, CSS, JavaScript, SQL, and PHP. Now your goal is to review my web technology project proposal and provide feedback and suggestions for the necessary improvements. Please ensure that feedback is insightful and encouraging. Please let me know if you have any questions before providing feedback.

The above example adheres to the structure of a prompt formula 3.1 by clearly outlining the task and providing all necessary information for a detailed review. It is crafted to optimize the feedback process, emphasizing constructive criticism and actionable suggestions that encourage both improvement and learning.

3.2 Evaluation and Results Analysis

This evaluation was designed to measure how effectively prompt engineering improves the relevance and usefulness of AI-generated feedback from the students' viewpoint. While prompt engineering techniques enable us to obtain specific responses, it remains unclear how beneficial this feedback truly is to students and in what areas it may fall short. Moreover, we aim to identify the types of feedback tasks for which ChatGPT feedback proves most usable, and those for which it may be insufficient. This analysis will help us understand the strengths and limitations of ChatGPT in providing better feedback to students for a Web Technology course.

To assess the effectiveness of the feedback generated by ChatGPT, we utilized data collected from the course from two different sources: 1) Teaching Evaluation; 2) Student ratings and insights on ChatGPT feedback through each checkpoint project report.

1. **Teaching Evaluation:**

   In two academic years (2022 and 2023), the web technologies course was taught using the same project-based learning module by Dr. Zheng Song. However, there was a significant difference in the approach to student feedback between the two iterations. In the first year, students received feedback from Instructors and TAs for their checkpoints. In contrast, the second year implemented a method whereby students were coached in prompt engineering to get feedback from ChatGPT, including the customization of prompts tailored to their needs. According to Table 3.1, which compares the teaching evaluations for these two years, there was a noticeable improvement in course satisfaction rates in the second year. This suggests that prompt engineering significantly enhances students' ability to receive feedback, thereby improving their overall learning experience.

2. **Student ratings and insights on ChatGPT feedback through each checkpoint project report:**

   We encouraged students to utilize ChatGPT for obtaining feedback on their project checkpoints, asking them to evaluate the feedback on a 1 to 5 scale and offer insights about it. Specifically, we requested that students detail their experiences, highlighting the advantages and disadvantages of receiving feedback from ChatGPT. Furthermore, at the course end, we incorporated a couple of questions into the post-course survey about prompt engineering and ChatGPT to assess their understanding of these concepts.

By comparing the teaching evaluations of these two consecutive years from Table 3.1 and figure [**?**], we can see that the second year's mean 4.56, median 5.0, and overall course satisfaction rate 4.56 are higher compared to the first year mean 4.24, median 4.0, and overall course satisfaction rate 4.4. This suggests that receiving feedback from ChatGPT had a significant impact on course learning.

Year	No. of Students	No. of Questions	Response Rate	Mean	Median	Course Rating
2022	56	13	48.21	4.24	4.0	4.4
2023	30	13	46.67	4.56	5.0	4.56

Table 3.1: Course Evaluation Summary

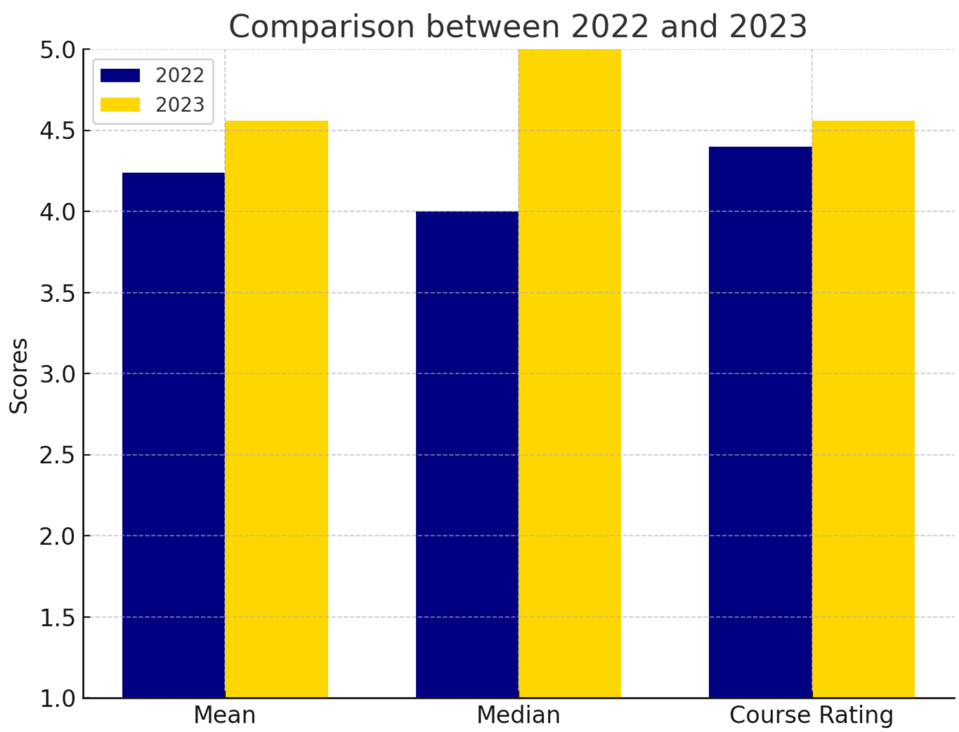


Figure 3.2: Teachinhg Evaluation Results

This analysis examines how students interacted with ChatGPT for feedback, both with and without pre-designed prompts. We compared the effectiveness of ChatGPT's feedback across different project checkpoints to assess its impact on the learning experience in our Web Technology course. Specifically, we explored how prompt engineering techniques helped students receive feedback, the tasks where ChatGPT's feedback proved most beneficial (e.g., project proposals, coding), and its limitations. This evaluation is crucial for optimizing ChatGPT's integration as a supportive tool in education.

From the analysis we found that students gave an average rating of 3 out of 5 to the feedback from ChatGPT when using pre-designed prompts and agreed that prompt engineering techniques are helpful for obtaining feedback 3.4. In contrast, feedback received without the use of pre-designed prompts was rated an average of 1 out of 5 as shown in the 3.3 The feedback from ChatGPT with pre-designed prompts was found to be somewhat more specific compared to when no prompts were used.

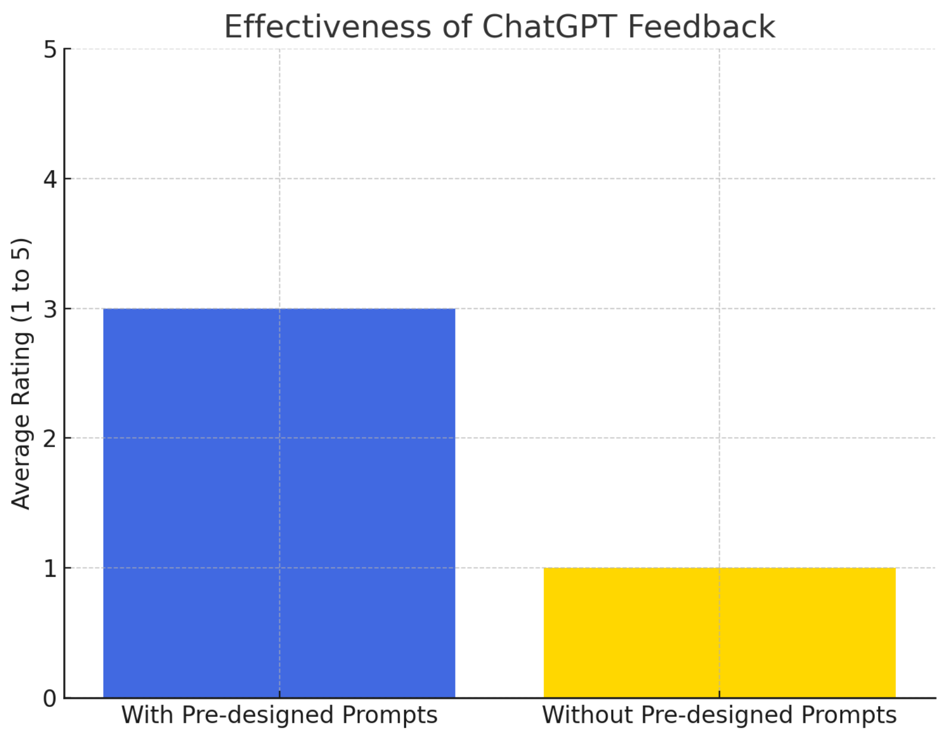


Figure 3.3: Student Ratings

Benefits and Limitations of ChatGPT Feedback: General purpose ChatGPT with pre-designed prompts offered some advantages. Those advantaged are given below:

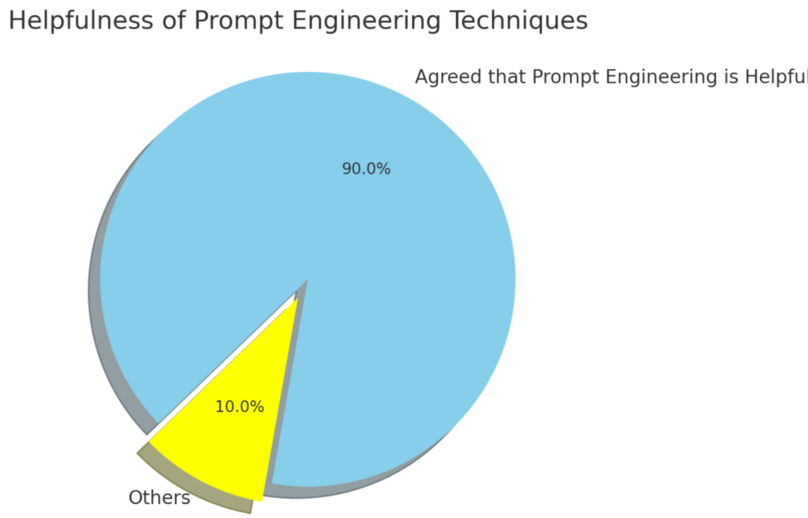


Figure 3.4: Student Ratings

- **UI/UX Design and Functionality**: ChatGPT offered some useful suggestions regarding user interface/user experience (UI/UX) design and functionalities.

- **Coding Assistance**: It aided in code debugging and provided assistance when students encountered coding issues.

- **Creative Ideas**: ChatGPT often provided a good starting point for generating creative ideas.

Despite these advantages, it still falls short in several areas, as highlighted by student evaluations:

Observation 1

1. Offering high level suggestions instead of focusing on problem area.

2. Providing suggestions that fall outside the intended scope.

3. Struggling to provide proper code review and validation.

While students acknowledged that prompt engineering helps them seek feedback from ChatGPT, they also identified several shortcomings. Consequently, we aimed to develop a customized tool designed to address these limitations. To achieve this, we created a Customized GAI tool for feedback provisioning that addresses all the issues highlighted in the observations listed in Chapter IV.

CHAPTER 4

Customizing GAI for for Feedback Provisioning

We developed an approach to enhance 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.

4.1 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.

4.1.1 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[28, 29]. 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:

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

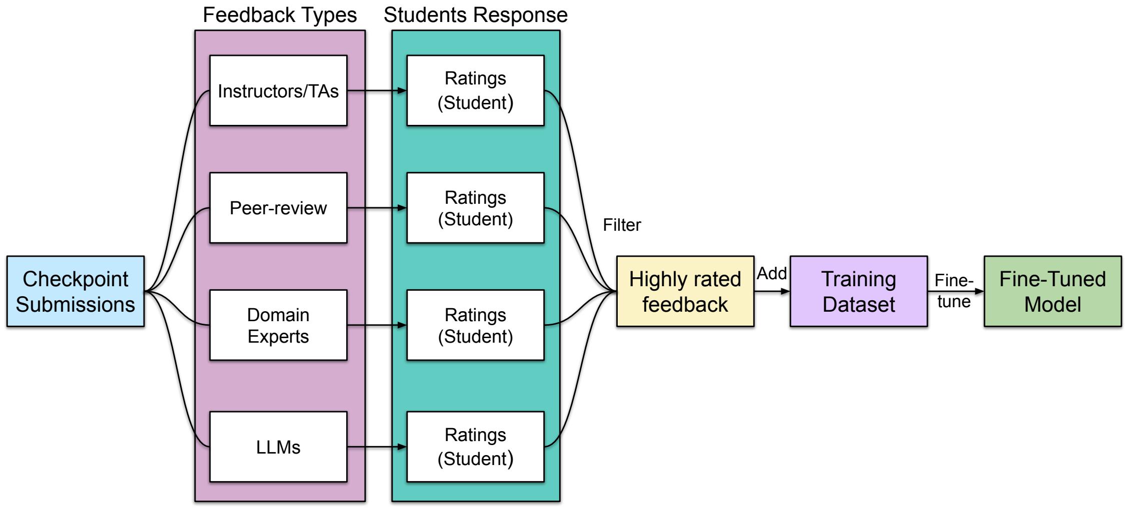


Figure 4.1: Steps involved in retaining valuable feedback

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. 4.1 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.

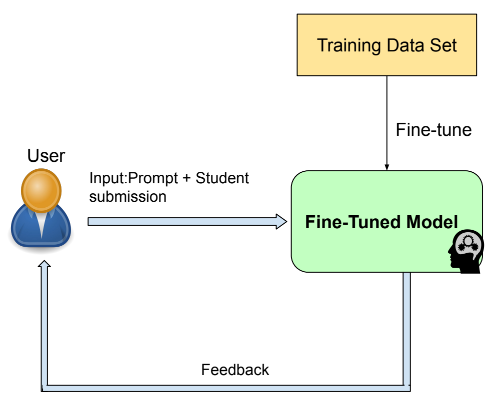


Figure 4.2: Fine-Tuned Method Workflow

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. 4.2 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.

4.1.2 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) method[30] 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[30, 31].

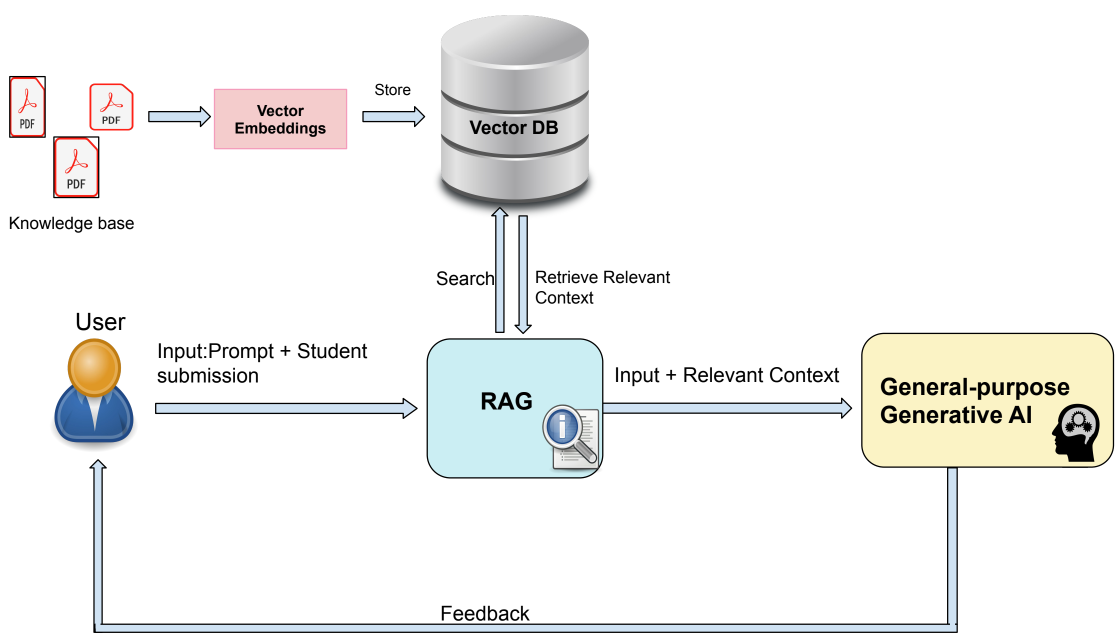


Figure 4.3: Additional Context Work Flow

Fig. 4.3 shows the workflow of this method. The relevant information is cut into smaller pieces and converted into word vectors by applying embedding [32], 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.

4.1.3 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.

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. 4.4. 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

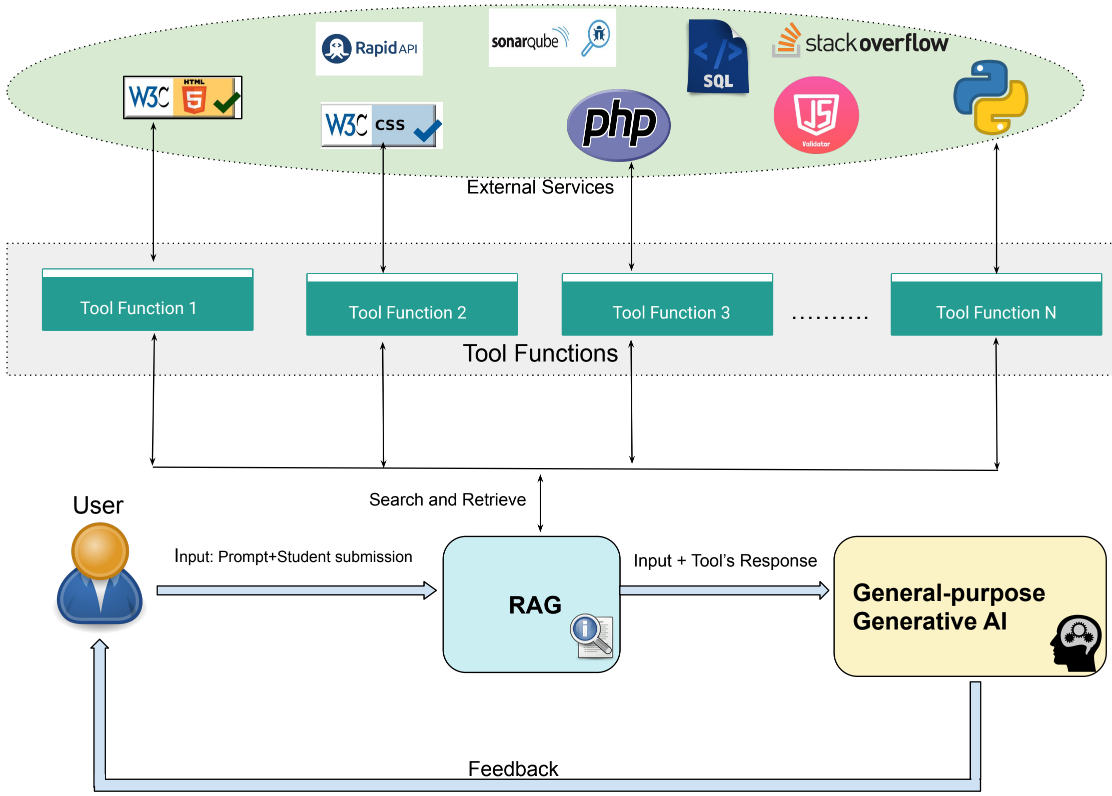


Figure 4.4: External Services Workflow

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.

4.2 Implementation

This section presents the implementation of the proposed methods outlined in Section 4.1. 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.

4.2.1 Fine-tuning ChatGPT

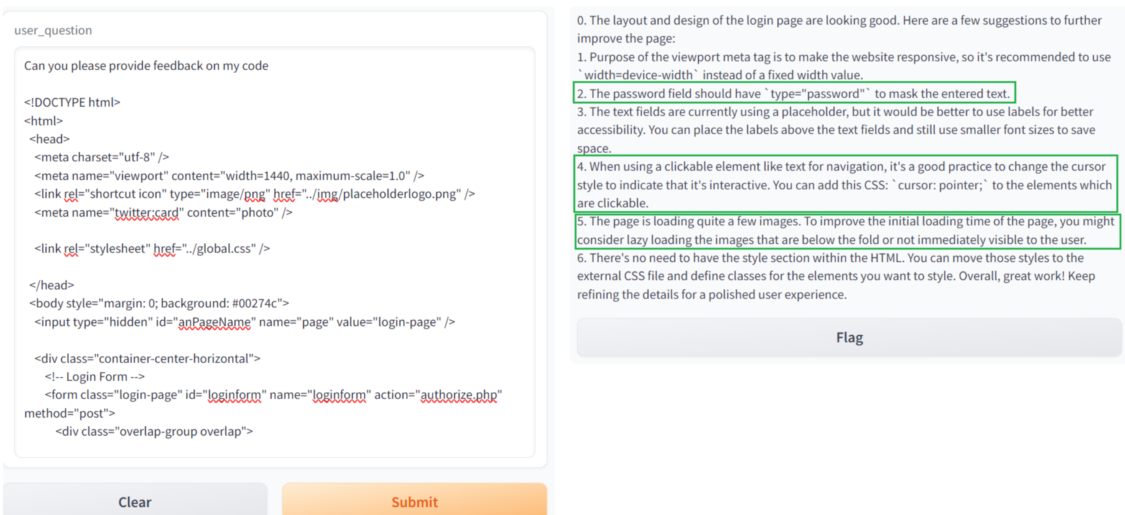


Figure 4.5: Feedback generated by a fine-tuned model on HTML codebase

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. 4.1, 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 model[**?**].

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. 4.5, 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.

4.2.2 Providing Relevant Information related to Students' Submission

To provide course-relevant materials as additional context, our tool implements Retrieval Augmented Generation (RAG) using the LangChain framework[33]. 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. 4.6, 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. 4.7 shows that the feedback generated by our tool mentions about these specific rubrics

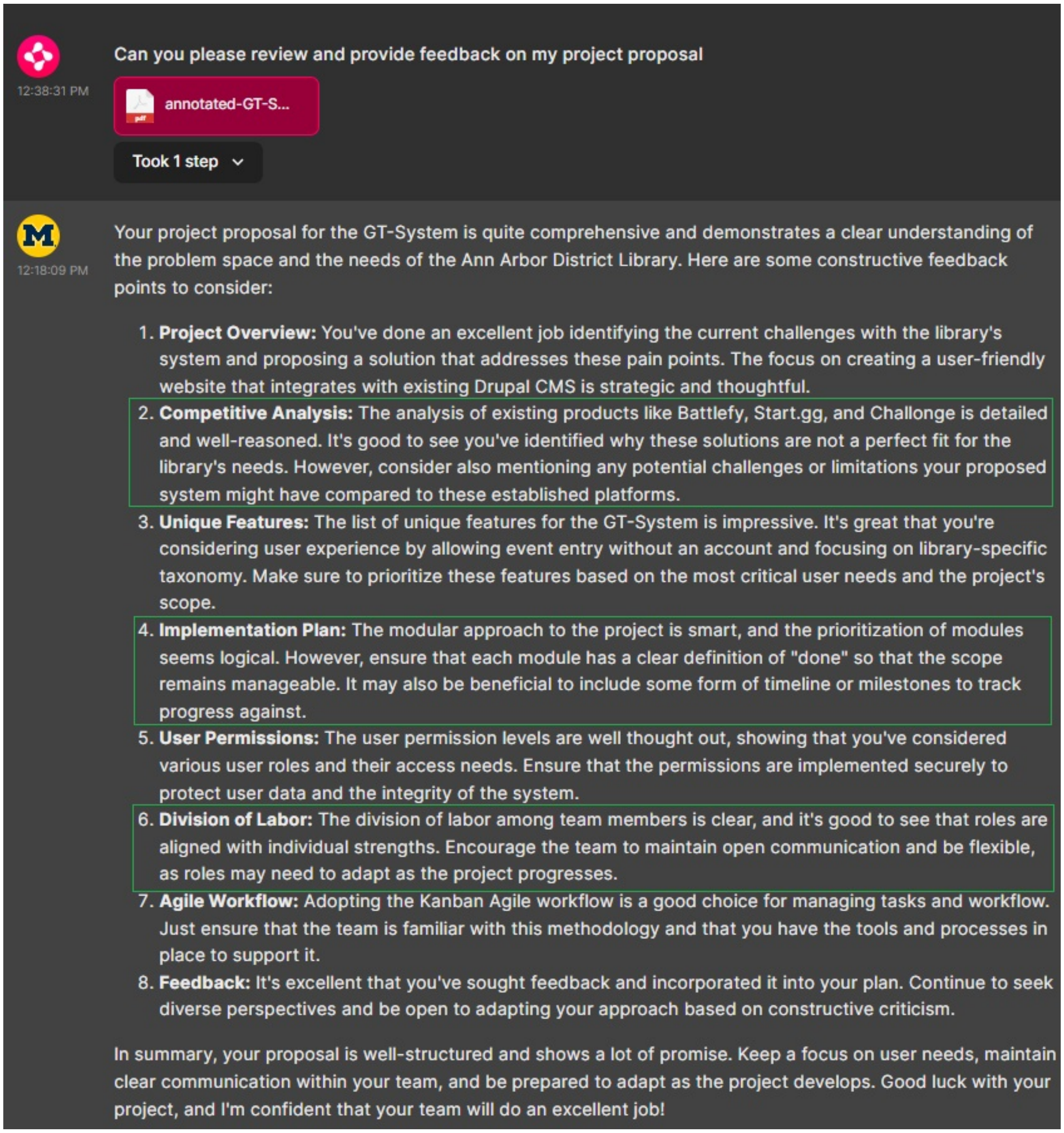


Figure 4.6: Feedback on project proposal using relevant context

points (competitive analysis, implementation plan, workload division, etc.), thus verifying that it can access such course-relevant information while generating feedback.

4.2.3 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

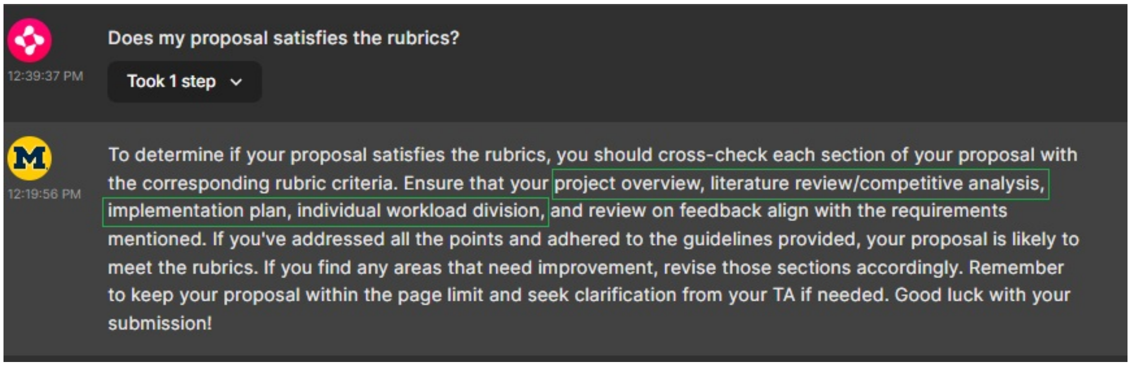
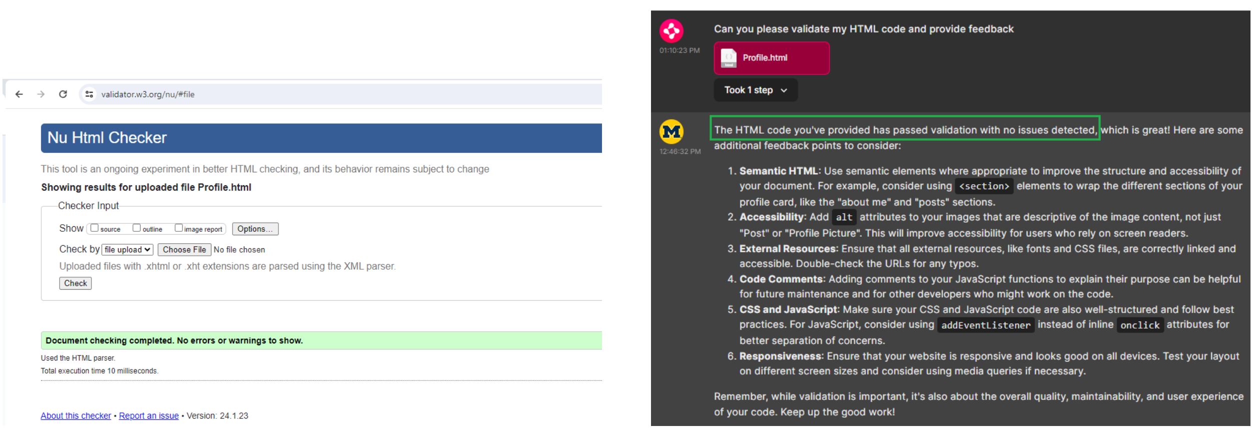


Figure 4.7: Course-relevant Additional Context (rubrics)

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.



(a) W3C Validator

(b) Response after incorporating W3C Validator

Figure 4.8: Integrating W3C Validator as External Service

We tested our tool against a HTML code validation task, and the response obtained is showcased in Fig. 4.8, where it is able to perform an HTML validation check using the HTML W3C validator

[1] https://validator.w3.org/
[2] http://sqllint.com/
[3] https://github.com/PHP-CS-Fixer/PHP-CS-Fixer/

(Fig. 4.8a) 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.

4.2.4 A Combined Framework

All three methods described earlier in Section 4.1 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.

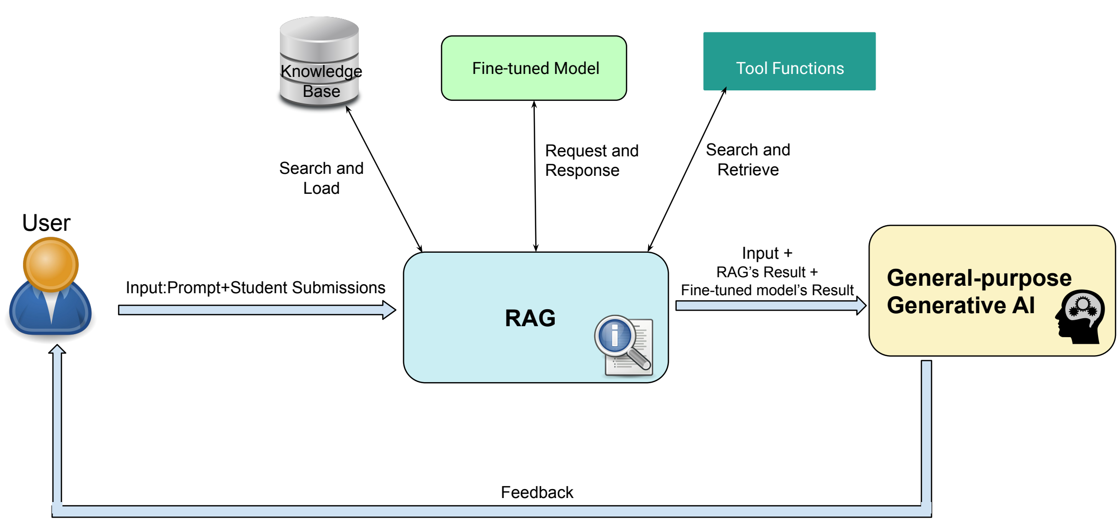


Figure 4.9: Combined Framework

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. 4.9. When a student submits a prompt along with their submissions, it is first routed to RAG, which acts as an entry point to 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 4.1, 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. 4.10, 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.

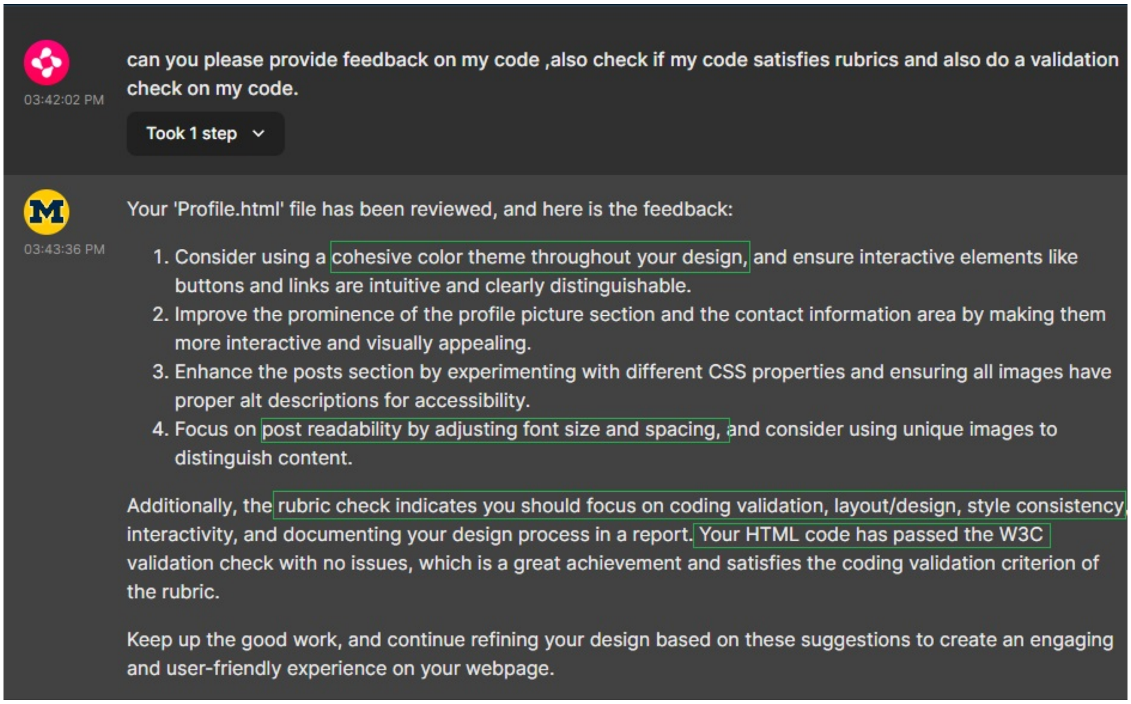


Figure 4.10: Feedback using (Fine-tuning + Additional Context + External Services)

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.

4.3 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.

4.3.1 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 4.1, resulting in a total of 18 different questions. Table 4.1 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	**Comparison of methods**
Q1	Asking for feedback on project proposal	1. General-Purpose ChatGPT (GPCG) 2. Additional Context (AC)
Q2	Asking if Q1 meets the rubrics requirements as a follow-up question.	
Q3	Asking for feedback on codebase	1. General-Purpose ChatGPT (GPCG) 2. Fine-tuning (FT) 3. Additional Context (AC)
Q4	Asking if Q3 meets the rubrics requirements as a follow-up question.	
Q5	Asking for code validation and feedback	1. General-Purpose ChatGPT (GPCG) 2. Fine-tuning (FT) 3. External Services as Tools (ET)
Q6	Asking for feedback on visual elements and UI designs	
Q7	Asking for feedback based on rubrics requirements and code validation	1. Fine-tuning (FT) + Additional Context (AC) 2. Fine-tuning (FT) + Additional Context (AC) + External Services as Tools (ET)

Table 4.1: Survey Questionnaire

4.3.2 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 4.11a, 4.11b, 4.11c, 4.11d, 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.

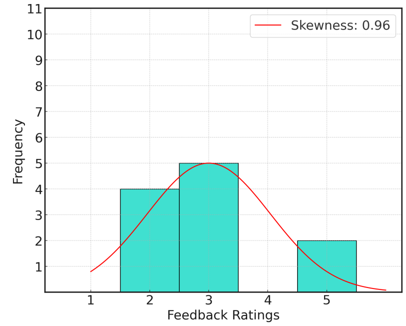
> **Observation 2**
>
> The Additional Context(AC) method, providing course-relevant information, is better for generating feedback on project proposals as it not only offers feedback but also responds to questions regarding proposal rubrics related to the course, which cannot be answered by ChatGPT(GPCG).

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 4.11e,4.11f,4.11g,4.11h,4.11i and 4.11j 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.

(a) Q1 - GPCG

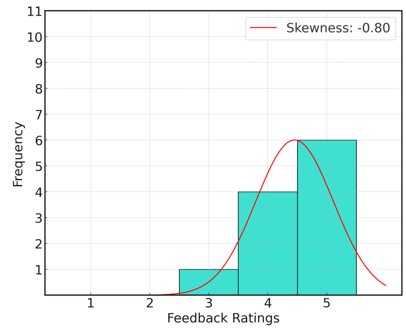
(b) Q1 - AC



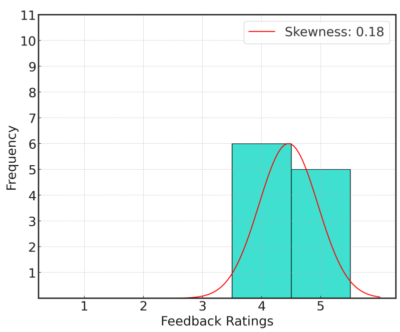
(c) Q2 - GPCG

(d) Q2 - AC

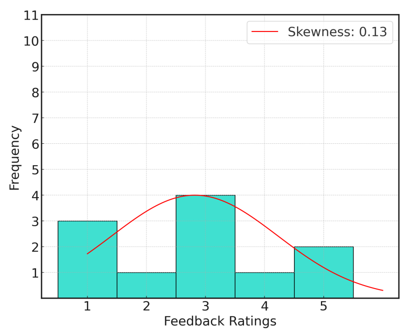
(e) Q3 - GPCG



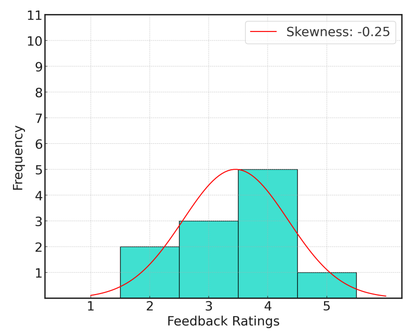
(f) Q3 - FT



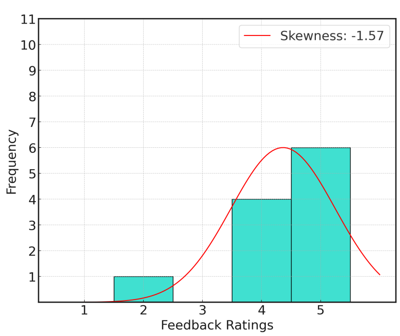
(g) Q3 - AC



(h) Q4 - GPCG

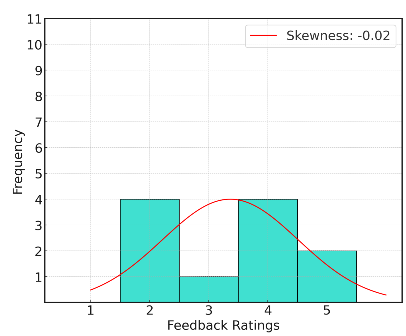


(i) Q4 - FT



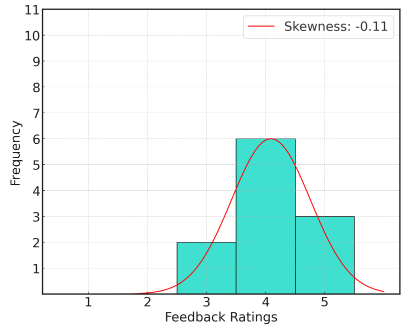
(j) Q4 - AC

(k) Q5 - GPCG

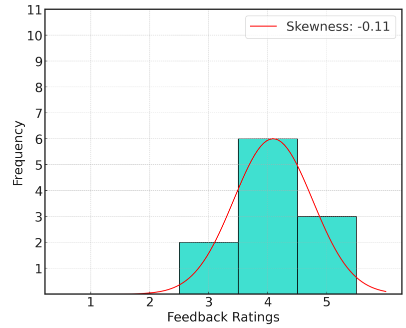


(l) Q5 - FT

(m) Q5 - ET

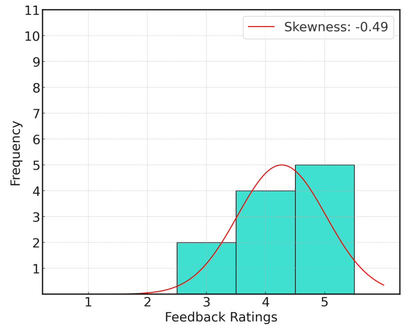


(n) Q6 - GPCG

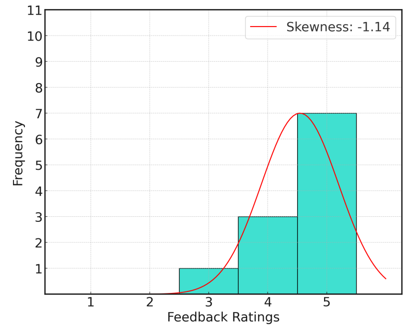


(o) Q6 - FT

(p) Q6 - ET



(q) Q7 - FT + AC



(r) Q7 - FT + AC + ET

Figure 4.11: Effectiveness of various feedback methods (Q1 - Q7)

Observation 3

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 4.11k,4.11l, and 4.11m 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 4

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 4.11n,4.11o, and 4.11p 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 5

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 4.11q and 4.11r 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 6

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.

CHAPTER 5

Conclusion and Future Directions

This paper presents a novel approach for enhancing a general-purpose AI model like ChatGPT for Project-Based Learning in computer science education. We successfully developed a specialized tool that leverages ChatGPT's capabilities and addresses its limitations in providing project feedback. While our initial evaluation showed promising results, further research is needed to explore its effectiveness in larger and more diverse settings. We believe this work has the potential to revolutionize PBL by facilitating broader adoption through automated feedback and a more dynamic learning experience for students.

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 from three different student projects, 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 it 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 students with sufficient background on it, so they consider it as a learning companion rather than completely relying on it, thus enhancing 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

APPENDIX

Relevant Paper Publications

During my journey as an MS student, I have had one paper published at the ASEE (2024 Annual Conference of the American Society for Engineering Education), one paper abstract submitted to FIE, and one paper under submission to a journal. I am the first author of all these submissions. The papers are listed below.

The paper titled **"Generative-AI Assisted Feedback Provisioning for Project-based Learning in CS Courses"** was published at the ASEE (2024 Annual Conference of the American Society for Engineering Education).

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 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.

Another paper titled **“A PBL-based Mini Course Module for Teaching Computer Science Students to Utilize Generative AI for Enhanced Learning“** abstract was submitted to the Frontiers in Education conference (FIE) 2024.

Abstract:

This research-to-practice paper introduces a mini-course module designed to teach computer science students how to interact more efficiently with Generative AI (GAI). The rapid rise of GAI is transforming education by providing students with easy access to knowledge and answers to their questions, acting as a personal tutor. Particularly in the field of computer science, where GAI can easily generate code based on specific requirements, many instructors struggle to prevent students from using tools like ChatGPT for completing assigned programming assignments and homework. However, we argue that 1) the use of GAI is inevitable, necessitating a redesign of courses so that students cannot merely rely on GAI without actual learning; and 2) students' learning can be enhanced if they learn to use GAI more effectively. In this paper, we demonstrate how we integrate Project-Based Learning to design the course module in a concise yet effective manner, which not only facilitates students' learning of GAI but also enriches their learning in relation to the host course where this mini-course module is embedded.

In particular, the goal of this module is to teach CS students: 1) the basic principles and workflow of GAI; 2) Prompt Engineering: how to craft questions to interact more effectively with GAI; and 3) Extending GAI: how to create interactive tools by training customized GAI models. Designed to be completed within two weeks, the mini-course module can easily be incorporated into host courses. In the first week, we offer lectures and interactive examples to introduce the basic knowledge of the three topics listed above. We also design in-class exercises where students can explore how to most effectively ask questions about topics taught in the host course. In the second week, students are encouraged to develop projects of their own choice, focusing on building a GAI-based tool related to subjects from the host course. This project-based approach immerses students in authentic challenges and provides the flexibility for personalized exploration. To reduce the learning curve for students, boilerplate code examples and selected project examples will be provided.

We plan to pilot this mini-course module in a graduate-level Artificial Intelligence course with 40 students in Winter 2024. The example project we will give to students involves training ChatGPT to simulate the output of various clustering algorithms. To assess the module's impact on student learning and engagement, we will conduct pre- and post-course surveys as well as student interviews. The results from these surveys and interviews will provide valuable insights and help us understand how the design of educational modules can be improved to effectively leverage and equip students with essential GAI competencies that foster student engagement and learning within a concise timeframe.

References

[1] L. Zhang and Y. Ma, "A study of the impact of project-based learning on student learning effects: a meta-analysis study," *Frontiers in Psychology*, vol. 14, 07 2023.

[2] D. Kokotsaki, V. Menzies, and A. Wiggins, "Project-based learning: A review of the literature," *Improving schools*, vol. 19, no. 3, pp. 267–277, 2016.

[3] R. Dookwah and G. Julien, "Motivation heightens and independent thinking deepens: Undergraduate students share their experiences of pbl while learning microeconomics." *Educational Research and Reviews*, vol. 17, no. 8, pp. 227–233, 2022.

[4] J. Hattie and H. Timperley, "The power of feedback," *Review of educational research*, vol. 77, no. 1, pp. 81–112, 2007.

[5] K. Aga, "Challenges and motivation for teachers transitioning to active learning spaces," *European Journal of Engineering Education*, pp. 1–23, 2023.

[6] J. Qadir, "Engineering education in the era of chatgpt: Promise and pitfalls of generative ai for education," in *2023 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2023, pp. 1–9.

[7] W. Dai, J. Lin, H. Jin, T. Li, Y.-S. Tsai, D. Gašević, and G. Chen, "Can large language models provide feedback to students? a case study on chatgpt," in *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2023, pp. 323–325.

[8] C. Zheng, K. Yuan, B. Guo, R. H. Mogavi, Z. Peng, S. Ma, and X. Ma, "Charting the future of ai in project-based learning: A co-design exploration with students," *arXiv preprint arXiv:2401.14915*, 2024.

[9] W. A. Friess and A. J. Goupee, "Using continuous peer evaluation in team-based engineering capstone projects: A case study," *IEEE Transactions on Education*, vol. 63, no. 2, pp. 82–87, 2020.

[10] K. Smith, N. Maynard, A. Berry, T. Stephenson, T. Spiteri, D. Corrigan, J. Mansfield, P. Ellerton, and T. Smith, "Principles of problem-based learning (pbl) in stem education: Using expert wisdom and research to frame educational practice," *Education Sciences*, vol. 12, no. 10, p. 728, 2022.

[11] D. Kokotsaki, V. Menzies, and A. Wiggins, "Project-based learning: A review of the literature," *Improving Schools*, vol. 19, no. 3, pp. 267–277, 2016.

[12] S. C. dos Santos, P. B. Reis, J. F. Reis, and F. Tavares, “Two decades of pbl in teaching computing: a systematic mapping study,” *IEEE transactions on education*, vol. 64, no. 3, pp. 233–244, 2020.

[13] J. P. Fernandes, R. Araújo, and M. Zenha-Rela, “Achieving scalability in project based learning through a low-code platform,” in *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, 2020, pp. 710–719.

[14] A. Jaber, K. Lechasseur, K. Mottakin, and Z. Song, “A study report in the web technologies course: What makes feedback effective for project-based learning?” in *2023 ASEE Annual Conference & Exposition*, 2023.

[15] J. Hattie and H. Timperley, “The power of feedback,” *Review of Educational Research*, vol. 77, no. 1, pp. 81–112, 2007. [Online]. Available: https://doi.org/10.3102/003465430298487

[16] A. C. Koenka and E. M. Anderman, “Personalized feedback as a strategy for improving motivation and performance among middle school students,” *Middle School Journal*, vol. 50, no. 5, pp. 15–22, 2019.

[17] B. de la Torre-Neches, M. Rubia-Avi, J. L. Aparicio-Herguedas, and J. Rodríguez-Medina, “Project-based learning: an analysis of cooperation and evaluation as the axes of its dynamic,” *Humanities and Social Sciences Communications*, vol. 7, no. 1, pp. 1–7, 2020.

[18] R. Al-Hammoud and O. Egbue, “The role of feedback in enhancing students’ learning experience: An evaluation of student perspectives and attitudes,” in *2023 ASEE Annual Conference & Exposition*, 2023.

[19] V. Razmov and S. Vlasseva, “Feedback techniques for project based courses,” in *2004 Annual Conference*, 2004, pp. 9–608.

[20] M. Abu-Ghuwaleh and R. Saffaf, “Integrating ai and nlp with project-based learning in stream education,” 2023.

[21] M. Montenegro-Rueda, J. Fernández-Cerero, J. M. Fernández-Batanero, and E. López-Meneses, “Impact of the implementation of chatgpt in education: A systematic review,” *Computers*, vol. 12, no. 8, 2023. [Online]. Available: https://www.mdpi.com/2073-431X/12/8/153

[22] V. Connolly and S. Watson, “Chatgpt and education.”

[23] I. Purnama, F. Edi, Syahrul, R. Agustin, and N. W. Pranoto, “Gpt chat integration in project based learning in learning: A systematic literature review,” *Jurnal Penelitian Pendidikan IPA*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:267231259

[24] S. Ali, P. Ravi, R. Williams, D. DiPaola, and C. Breazeal, “Constructing dreams using generative ai,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 268–23 275.

[25] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," 2023.

[26] S. Ekin, "Prompt engineering for chatgpt: A quick guide to techniques, tips, and best practices," May 2023. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.22683919.v2

[27] D. C. Schmidt, J. Spencer-Smith, Q. Fu, and J. White, "Towards a catalog of prompt patterns to enhance the discipline of prompt engineering," 2023.

[28] T. Nguyen-Mau, A.-C. Le, D.-H. Pham, and V.-N. Huynh, "An information fusion based approach to context-based fine-tuning of gpt models," *Information Fusion*, vol. 104, p. 102202, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523005183

[29] "Patent claim generation by fine-tuning openai gpt-2," *World Patent Information*, vol. 62, p. 101983, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0172219019300766

[30] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, "A survey on retrieval-augmented text generation," *arXiv preprint arXiv:2202.01110*, 2022.

[31] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[32] K. Matsuda and I. Frank, "Langchain unleashed: Advancing education beyond chatgpt's limits."

[33] O. Topsakal and T. C. 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*, 2023, pp. 10–12.

[34] C. K. Lo, "What is the impact of chatgpt on education? a rapid review of the literature," *Education Sciences*, vol. 13, no. 4, p. 410, 2023.

[35] K. Buffardi and S. H. Edwards, "Reconsidering automated feedback: A test-driven approach," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, pp. 416–420.

[36] O. Johnson, O. Alyasiri, D. Akhtom, and O. Johnson, "Image analysis through the lens of chatgpt-4," *Journal of Applied Artificial Intelligence*, vol. 4, no. 2, 2023.

[37] M. Bucur, "Exploring large language models and retrieval augmented generation for automated form filling," July 2023. [Online]. Available: http://essay.utwente.nl/96259/

[38] "Summary of chatgpt-related research and perspective towards the future of large language models," *Meta-Radiology*, vol. 1, no. 2, p. 100017, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2950162823000176

[39] L. Boels, A. Bakker, W. Van Dooren, and P. Drijvers, "Conceptual difficulties when interpreting histograms: A review," *Educational Research Review*, vol. 28, p. 100291, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1747938X18304615

[40] H. M. Hammouri, R. T. Sabo, R. Alsaadawi, and K. A. Kheirallah, "Handling skewed data: A comparison of two popular methods," *Applied Sciences*, vol. 10, no. 18, p. 6247, 2020.

[41] D. W. Scott, "Histogram," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 44–48, 2010.

[42] B. Cheang, A. Kurnia, S. I. C. IOPT, A. Lim, and W.-C. Oon, "Automated grading of programming assignments."

[43] N. Shin, J. Bowers, J. Krajcik, and D. Damelin, "Promoting computational thinking through project-based learning," *Disciplinary and Interdisciplinary Science Education Research*, vol. 3, no. 1, pp. 1–15, 2021.

[44] S. Draft, "Computer science curricula 2013," *ACM and IEEE Computer Society, Incorporated: New York, NY, USA*, 2013.

[45] S.-H. Chang, L.-C. Yu, Y.-K. Kuo, Y.-T. Mai, and J.-D. Chen, "Applying online peer assessment with total quality management to elevate project-based learning performance," *Journal of Baltic Science Education*, vol. 14, no. 3, p. 379, 2015.

[46] A. Jaime, J. M. Blanco, C. Domínguez, A. Sánchez, J. Heras, and I. Usandizaga, "Spiral and project-based learning with peer assessment in a computer science project management course," *Journal of Science Education and Technology*, vol. 25, no. 3, pp. 439–449, 2016.

[47] M. Dahmen, L. Quezada, M. Alfaro, G. Fuertes, C. Aballay, and M. Vargas, "Teaching artificial intelligence using project based learning," EasyChair, Tech. Rep., 2020.

[48] A. Martínez-Monés, E. Gómez-Sánchez, Y. A. Dimitriadis, I. M. Jorrín-Abellán, B. Rubia-Avi, and G. Vega-Gorgojo, "Multiple case studies to enhance project-based learning in a computer architecture course," *IEEE Transactions on Education*, vol. 48, no. 3, pp. 482–489, 2005.

[49] P. J. Frontera and E. J. Rodríguez-Seda, "Network attacks on cyber–physical systems project-based learning activity," *IEEE Transactions on Education*, vol. 64, no. 2, pp. 110–116, 2020.

[50] D. Kundra and A. Sureka, "An experience report on teaching compiler design concepts using case-based and project-based learning approaches," in *2016 IEEE Eighth International Conference on Technology for Education (T4E)*. IEEE, 2016, pp. 216–219.

[51] W. Wahyudi, "The effectiveness of sharing blended project based learning (sbpbl) model implementation in operating system course," *International Journal of Emerging Technologies in Learning (IJET)*, vol. 15, no. 5, pp. 202–211, 2020.

[52] S. Gharbi, H. Bellakhdar, and S. E. Mrabet, "Project based learning in business intelligence with intervention of companies," in *2015 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2015, pp. 384–387.

[53] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, “Using project-based-learning in a mobile application development course—an experience report,” *Journal of Visual Languages & Computing*, vol. 31, pp. 196–205, 2015.

[54] D. Anitha, C. Jeyamala, and D. Kavitha, “Assessing and enhancing creativity in a laboratory course with project based learning,” *Journal of Engineering Education Transformations*, vol. 32, no. 2, pp. 67–74, 2018.

[55] V. V. Riabov and B. J. Higgs, “Running a computer security course: Challenges, tools, and projects,” *The Journal of Computing Sciences in Colleges*, p. 245, 2010.

[56] E. Martí, D. Gil, and C. Julià, “A pbl experience in the teaching of computer graphics,” in *Computer Graphics Forum*, vol. 25, no. 1. Wiley Online Library, 2006, pp. 95–103.

[57] V. S. Rathore, K. Rajawat, and S. Maheshwari, “Impact of open-ended project-based collaborative learning model for teaching undergraduate-level database management systems,” in *Third International Congress on Information and Communication Technology*. Springer, 2019, pp. 317–324.

[58] D. Kammer, E. Stoll, and A. Urban, “Experience of teaching data visualization using project-based learning,” *arXiv preprint arXiv:2111.04428*, 2021.

[59] F. Suryana, N. Jalinus, R. Rahmad, and R. Efendi, “Cooperative project based learning models in programming languages: A proposed,” *Int. J. Adv. Sci. Technol*, vol. 29, no. 06, pp. 1876–1886, 2020.

[60] B. Pérez and Á. L. Rubio, “A project-based learning approach for enhancing learning skills and motivation in software engineering,” in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 309–315.

[61] D. Delgado, A. Velasco, J. Aponte, and A. Marcus, “Evolving a project-based software engineering course: A case study,” in *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2017, pp. 77–86.

[62] R. Brüngel, J. Rückert, and C. M. Friedrich, “Project-based learning in a machine learning course with differentiated industrial projects for various computer science master programs,” in *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2020, pp. 1–5.

[63] M. L. Fioravanti, B. Sena, L. N. Paschoal, L. R. Silva, A. P. Allian, E. Y. Nakagawa, S. R. Souza, S. Isotani, and E. F. Barbosa, “Integrating project based learning and project management for software engineering teaching: An experience report,” in *Proceedings of the 49th ACM technical symposium on computer science education*, 2018, pp. 806–811.

[64] B. Opitz, N. K. Ferdinand, and A. Mecklinger, “Timing matters: the impact of immediate and delayed feedback on artificial language learning,” *Frontiers in human neuroscience*, vol. 5, p. 8, 2011.

[65] T. P. Cabre, M. T. Cairol, D. F. Calafell, M. T. Ribes, and J. P. Roca, "Project-based learning example: controlling an educational robotic arm with computer vision," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 8, no. 3, pp. 135–142, 2013.

[66] D. Geronimo, J. Serrat, A. M. Lopez, and R. Baldrich, "Traffic sign recognition for computer vision project-based learning," *IEEE transactions on education*, vol. 56, no. 3, pp. 364–371, 2013.

[67] S. Khorbotly, "A project-based learning approach to teaching computer vision at the undergraduate level," in *2015 ASEE Annual Conference & Exposition*, 2015, pp. 26–91.

[68] A. Stahlbauer, C. Frädrich, and G. Fraser, "Verified from scratch: program analysis for learners' programs," in *IEEE/ACM ASE'20*, 2020, pp. 150–162.

[69] T. Ito, M. S. Tanaka, M. Shin, K. Miyazaki *et al.*, "The online pbl (project-based learning) education system using ai (artificial intelligence)," in *E&PDE'21*, 2021.

[70] T. Kojima, S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *ArXiv*, vol. abs/2205.11916, 2022.

[71] H. Ma, C. Zhang, Y. Bian, L. Liu, Z. Zhang, P. Zhao, S. Zhang, H. Fu, Q. Hu, and B. Wu, "Fairness-guided few-shot prompting for large language models," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 43 136–43 155. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/8678da90126aa58326b2fc0254b33a8c-Paper-Conference.pdf

[72] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.