A MOBILE APPLICATION PLATFORM THAT SERVES AS A WRITING WORKSHOP FOR NARRATIVE EDITING AND DISCUSSIONS USING CHATGPT AS AN AI ASSISTANT

Meilin Shen [1], Bobby Nguyen [2]

[1] Sage Hill School, 20402 Newport Coast Dr, Newport Coast, CA 92657
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This paper presents a mobile application designed to improve the peer review and feedback process for student writers using a combination of community discussion tools and AI-guided support. The platform addresses challenges in receiving constructive feedback by offering structured peer engagement, instructional resources, and AI-generated suggestions through a guided prompt system. The app was developed using Flutter and Firebase, with components for user-authored drafts, a public feedback forum, and an AI Assistant [11]. Key challenges included ensuring feedback quality and controlling AI behavior, both of which were addressed through surveys, structured inputs, and testing. Experiments evaluated AI adherence to its role and system response times, confirming that prompt structure improves behavior and that system performance is impacted by network conditions. Compared to other scholarly methods, this solution emphasizes guided autonomy, trust calibration, and user-centered design [12]. Ultimately, the application enhances student learning, feedback exchange, and writing improvement while preventing overreliance on automated editing.

KEYWORDS

Writing, Editing, Narrative Editing, Discussion, Feedback

1. INTRODUCTION

In the mid-seventeenth century, the Royal Society of London introduced peer review, establishing a cornerstone of modern academic and scientific writing. Although peer review gained widespread recognition in the latter half of the twentieth century, its brief history belies its significant impact on publishing houses, newspapers, and scientific journals. In academia, peer dialogue has been shown to enhance students' comprehension and adaptability to feedback, contrasting with direct edits from instructors, which often correlate with lower student performance (van den Berg et al. 2006) [1]. Peer feedback offers numerous advantages, such as reducing teacher workloads and improving students' critical thinking and communication skills. Nicol and Macfarlane-Dick (2006) argue that effective feedback is essential for self-regulated learning, which peer review can facilitate when conducted systematically. However, implementing peer review in high school academic literature presents unique challenges and potential pitfalls. Research indicates a positive correlation between constructive feedback and improved student writing scores. Despite this, students frequently report inconsistent, untimely, and incomplete feedback, while teachers highlight the poor application of the advice provided.

David C. Wyld et al. (Eds): CSIT, AMLA, IPDCA, NLPA, AIS, IPPR, SPTM – 2025
pp. 135-145, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151410

Students often express skepticism about peer feedback's value due to the varied capabilities of their peers—some offer valuable, constructive feedback. In contrast, others lack the skills to provide high-quality input [2]. Surveys reveal significant student dissatisfaction with the quality of feedback received, emphasizing the need for instruction on constructive feedback principles. This underscores the critical need for an engaging, accessible, and systematic platform for providing and interpreting feedback to maximize its effectiveness in improving student writing.
Several methodologies were examined to address issues in human-AI collaboration. One study revealed that users often over-rely on AI when it performs well in unrelated tasks, highlighting a behavioral bias. While insightful, it lacked direct solutions, which our project improves upon by using structured prompts to guide user input and reduce over-dependence. Another study identified that ChatGPT's accurate responses come at the cost of slow response times, especially for complex tasks. Our approach resolves this by limiting ambiguity through guided prompts, improving speed and efficiency. A third methodology proposed a trust-based framework for AI collaboration, emphasizing dynamic trust calibration. Our application puts this into practice by embedding trust checkpoints in the user experience, reinforcing AI's supportive, not editorial role and maintaining user control.

The proposed solution entails establishing an online community for student writers, integrating interactive discussion features, instructional guides on effective feedback from English teachers, and generative AI conversational support [3]. An online platform offers greater accessibility than in-person settings, allowing a more significant, diverse sample size and a dedicated community of writers. This broad accessibility helps students identify feedback trends and build confidence, while smaller, in-person groups may foster doubt and impede revisions. Moreover, online platforms enable asynchronous communication, accommodating different schedules and learning paces. Students providing feedback must be trained in constructive criticism principles to mitigate carelessness and subjectivity. This platform offers a unified space for student writers seeking editing feedback. A survey feature requires student editors to answer systematically designed questions addressing English grammar, vocabulary, and structural aspects. This survey efficiently identifies feedback trends and enhances the basic understanding and composition of writing before students provide subjective feedback. The discussion board simulates a round-table discussion, allowing students to specify areas where they seek feedback. Peers can then provide detailed, specific suggestions for improvement, helping students gauge input resonance with a broader audience. The Learning Center focuses on guidelines for effective constructive feedback, featuring articles by English teachers and students who have researched scientifically proven feedback methods. An online platform also facilitates the archiving and retrieval of feedback, allowing students to revisit and reflect on past critiques, which is less feasible in in-person settings. The Artificial Intelligence assistance page has a list of four pre-programmed questions for the user to answer. The AI will then respond to the user's request for advice after reading the user's inquiries.

This paper conducted two experiments to evaluate blind spots in the application. The first experiment tested the AI Assistant's ability to remain within its intended role of offering guidance rather than performing direct edits. Structured prompts led to 100% adherence, while unstructured prompts caused significant deviation [4]. This confirmed that pre-programmed guidance questions are essential in maintaining user autonomy. The second experiment assessed the efficiency of client-server communication under various conditions. Response times were measured based on changes in internet quality and server load. Results showed that both poor connectivity and high usage significantly slowed response times, with average latency ranging from 1,800 ms to 7,200 ms. These findings emphasize the importance of controlling user input for accurate AI behavior and optimizing backend performance for user experience. Together, these tests validated the effectiveness of key system components and revealed areas where structured design and performance monitoring can further enhance platform functionality.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Teacher Survey

One of this program's primary objectives is to improve students' skills for giving informative, helpful, and constructive feedback. To create a guideline for advice and tips on how to effectively communicate comments and suggestions, a survey and open-answer questionnaire was sent out to ten English teachers. Questions in the survey included common hurdles in each stage of creative writing, how they could be combatted, mindful topics to consider when giving feedback, and how AI assistance in writing can be used without hindering the process of independent critical thinking. When gathering this information for the Learning Center, the main challenge was writing a guideline that followed the trend of advice from the surveys while distinguishing potential biases. Due to writing evaluation being a relatively subjective process, it was difficult to encompass the advice of all survey participants' emphasis on unique issues.

2.2. AI

The artificial intelligence model significantly influences this application's operational efficacy. One issue concerns programming AI to provide information strictly pertinent to the application's functionalities. Another challenge lies in effectively balancing AI capabilities with human expertise to foster a collaborative editing and feedback platform. The solution that was implemented to the app was the creation of a list of four questions regarding the current progress of the user's writing and the improvements they hope to make. This purpose of the pre-written list of questions will ensure that the AI will not impede on the diversity nor reduce accountability of the user's writing. Once the user answers these questions concisely under a 250 word limit, the AI will give straightforward recommendations to the user regarding the user's needs without directly editing the user's writing work.

2.3. User Account Creation Process

The primary challenge in programming the user account creation process using Firebase was generating a unique profile for a first-time user. The createUser function is needed to create a unique user ID, username, initialize an empty bio, profile picture, idea hub, published hub, and comments list. To generate a random username for each first-time user, I created two separate lists: one containing adjectives and the other containing writing genres. By combining one adjective and one writing genre through string interpolation, I constructed a distinctive username for every user. I implemented a verification process to ensure that only one copy of the new user account was created. Before creating a new entry, I verified the presence of the account's unique key in the shared preferences to check whether the account already existed on the user's device. If the account was not saved, I would create a user; if it was saved, I would allow the user to change their username, bio, and profile picture.

3. SOLUTION

This program consists of three main components: the user's private idea, the public browsing, and the AI assistance [5]. Upon creating an account, users can post their drafts via the draft screen and store their rough drafts in the idea screen. Once the writing is ready for publication, users can make their work publicly accessible, where it can be viewed by others through the published works screen. The published piece will also be updated on the ideas page, ensuring

consistency. Readers can select a piece to read, complete a feedback survey, and participate in feedback discussions. The author of the piece has exclusive access to the survey results, while the feedback discussion is publicly visible. To contribute to the feedback discussion, readers can use the expanded app bar to type comments and like others' comments, fostering an interactive community. The resource centers include the Learning Center and AI Assistance. The Learning Center features articles on providing effective feedback and writing advice, accessible to all users. AI Assistance offers additional support for writing and feedback processes by tailoring its assistance based on the specific user and task.

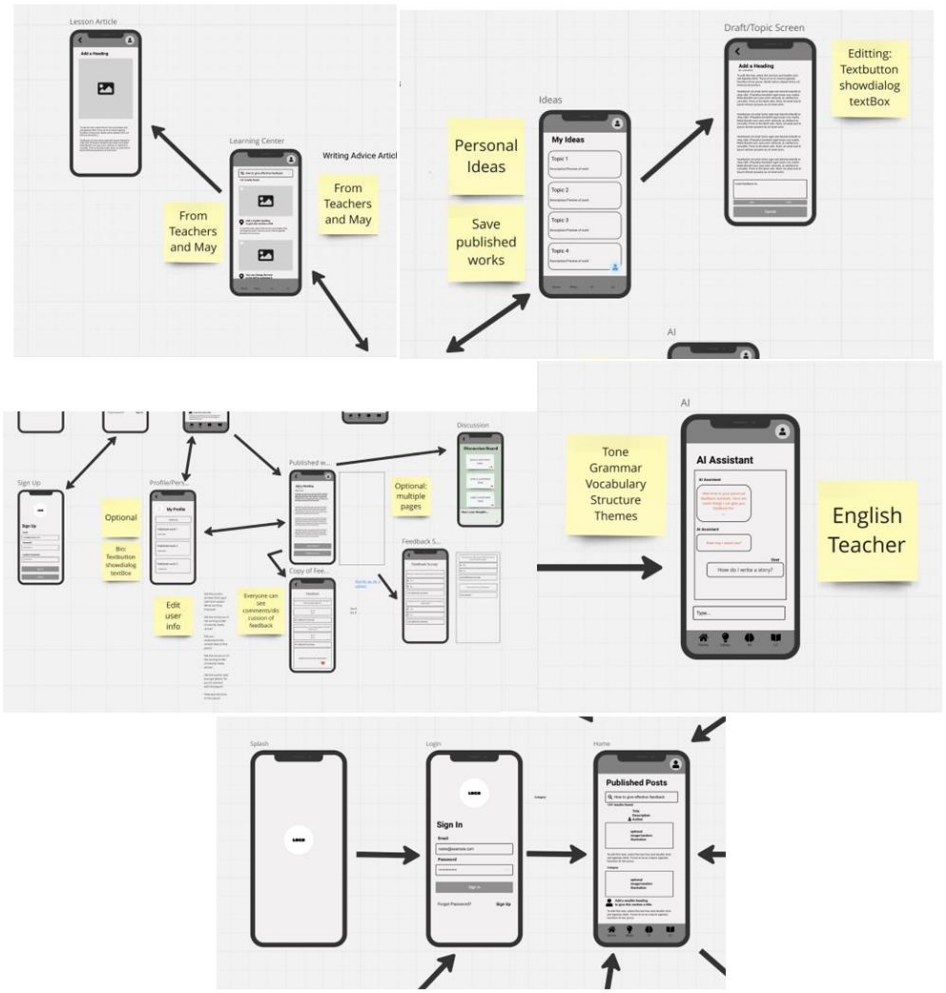


Figure 1. Overview of the solution

The public browsing field allows the user to scroll through the posts of all other users on the app and access them by clicking on the preview box. Once the user clicks on the post preview, the user will enter the full-length post. At the end of the post, there is the feedback and discussion accessor which holds the feedback questions list and discussion board. The viewer of the post can answer the questions and post a discussion, while the author of the post can view the answers and comments.

Figure 2. Screenshot of published posts

```
List search(fullList) {
  final tempList = [];
  if (searchText != '') {
    for (int i = 0; i < fullList.length; i++) {
      bool titleName = fullList[i]['title']
          .toLowerCase()
          .contains(searchText.toLowerCase());
      bool authorName = fullList[i]['username']
          .toLowerCase()
          .contains(searchText.toLowerCase());
      bool storyWord = fullList[i]['story']
          .toLowerCase()
          .contains(searchText.toLowerCase());
      if (titleName || authorName || storyWord) {
        tempList.add(fullList[i]);
      }
    }
  } else {
    return fullList;
  }

  return tempList;
}
```

Figure 3. Screenshot of code 1

This code directs the search element in the public works published screen by sifting through the full list of published posts for text that contains the String submitted by the user [6]. If the text body is not empty, the search function runs through the characters in the title, username, and body text of each published post until it matches the searched string.

The private ideas screen holds all the drafts of the user's work. This includes the already published work and works that the user is in the process of editing. The user can access each idea box, edit their work, and publish it at the bottom of the screen. Each draft page includes the title, author name, date, and body of text.

Figure 4. Screenshot of ideas

```
String shortContent(String body) {
  String tempSentence = "";
  int counter = 0;
  String tempChar = body[counter];

  while (counter < body.length - 1 && counter < 121) {
    tempSentence += tempChar;
    counter++;
    tempChar = body[counter];
  }

  return tempSentence;
}
```

Figure 5. Screenshot of code 2

This code segment shortens the full body text of the user's draft to a maximum of 120 characters before it is displayed by the preview text box. The while loop appends characters from the full body text to a new string, continuing until the text reaches 120 characters or stops early if the text is shorter than 120 characters.

The AI Assistant and Learning Center are resources that promote self-sufficiency outside of the social network screens. The AI Assistant reads through users' answers and provides open-ended suggestions that reflect the advice on the Learning Center [7]. The Learning Center holds articles and guidelines with research and teacher advice on feedback-giving and writing improvement.

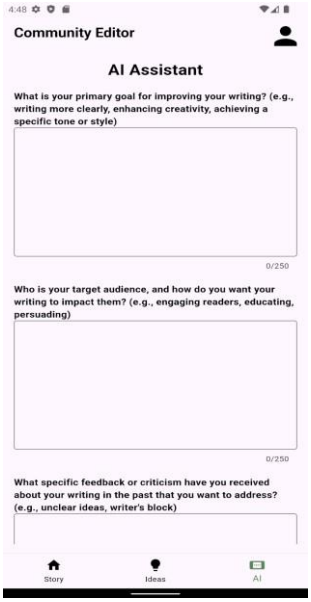


Figure 6. Screenshot of AI assistant

```
Future<void> handleMessage() async {
  List<String> questionList = [];
  List<String> answerList = [];
  for (int i = 0; i < 4; i++) {
    if (userResponses[i] != '') {
      questionList.add(questions[i]);
      answerList.add(userResponses[i]);
    }
  }

  String instructionPrompt =
      "You are an experienced author known for providing insightful and practical ad
      "Here are the questions they are answering: \n${questionList.join('\n')}"
      "Keep your advice short, concise, and to the point.";
  String userPrompt =
      "These are the user's answers: \n${answerList.join('\n')}";

  final request = ChatCompleteText(
    messages: [
      {
        'role': Role.system.name,
        'content': instructionPrompt,
      },
      {
        'role': Role.user.name,
        'content': userPrompt,
      },
    ],
    temperature: 0.2,
    maxToken: 1500,
    model: Gpt4ChatModel(), // Verify this model name
  ); // ChatCompleteText

  try {
    ChatCTResponse? response =
        await openAI.onChatCompletion(request: request);
    if (response != null && response.choices.isNotEmpty) {
      setState(() {
        reply = response.choices.first.message!.content.trim();
      });
    }
  } catch (e) {
    print("Error: ${e.toString()}");
  }
}
```

Figure 7. Screenshot of code 3

This code runs the AI Assistance component, beginning with a for-loop that checks if the user has entered a response to the pre-programmed questions, and then adds their input onto a list of answers [15]. Next, the code directs the AI to concisely answer the user's questions while keeping advice open ended without directly changing the user's work. In the preceding lines, the instruction prompt tells the AI how to respond, and the user prompt tells the AI what to respond to. The try function is attempting to catch any errors that would occur in the AI's process of responding. If an error is caught, an error is printed out instead of the AI advice and the program will refresh.

4. EXPERIMENT

4.1. Experiment 1

A potential blind spot is the extent of AI assistance, which must be carefully controlled. This is essential to prevent AI from undermining user independence and preserving the authenticity of peer feedback.

To evaluate this concern, I conducted a controlled experiment by deliberately attempting to push the AI beyond its intended supportive role. The AI Assistant is designed to guide users through self-reflection via four structured questions before offering writing suggestions. First, I followed this standard procedure to establish a baseline. Then, I bypassed the structured inputs and submitted direct commands such as “rewrite this paragraph,” “fix my grammar,” and “rephrase this sentence for clarity.” I analyzed whether the AI adhered to its designed constraint, providing only general guidance, or crossed into direct editing.

To simulate realistic user behavior, I copied sample drafts into the prompt window and requested edits. I repeated each test five times, assigning a percentage score to each attempt, based on how strictly the AI avoided making direct changes. For instance, a 100% score meant the AI offered only suggestions, while a 0% meant it edited text directly without constraint. This design helps measure how effectively the AI maintains its role as a feedback facilitator rather than a ghostwriter. The goal was to confirm that the structured questions effectively limited overreliance on AI and preserved student ownership of writing.

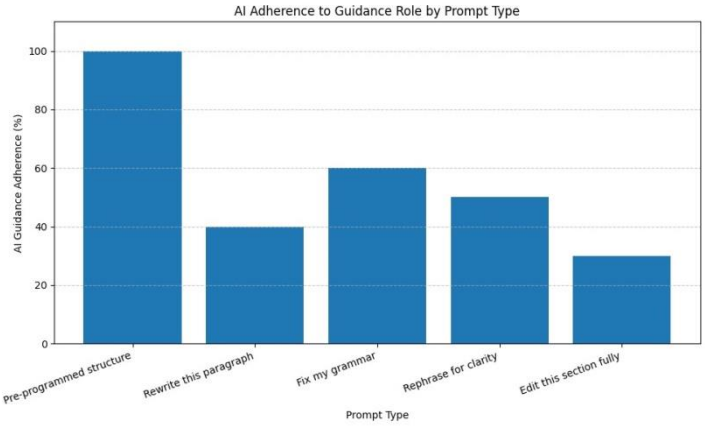


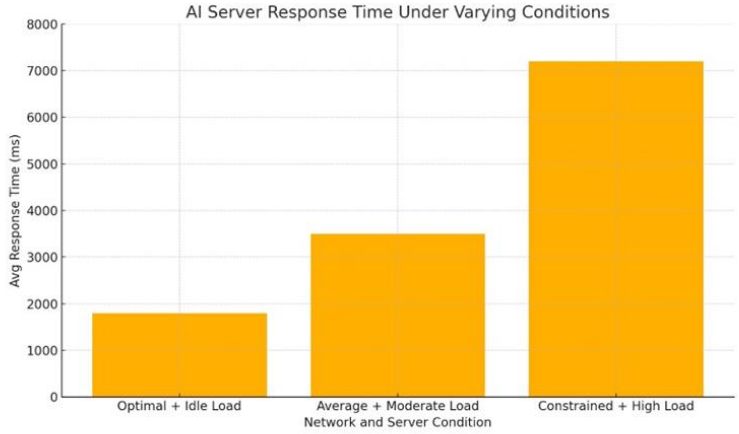
Figure 8. Figure of experiment 1

The AI maintained its guidance-only role perfectly (100%) when used with the intended preprogrammed structure. However, performance declined when directly prompted with editing tasks. The lowest adherence (30%) was seen when users explicitly asked for full edits. The average adherence across unstructured prompts was 45%, while the median was 50%. This deviation indicates that the AI sometimes interprets direct prompts as permission to edit, especially when language is strong or specific. These results highlight the importance of structured prompts in preventing misuse. The strongest influence on AI behavior was the format and clarity of the input. By standardizing user prompts, the app can significantly reduce unintended editing behavior. Future improvements might include hard-coded restrictions within the AI prompt template and real-time warnings if user prompts fall outside intended use. This ensures user autonomy remains intact and aligns the AI’s role strictly with guided support rather than content creation.

4.2. Experiment 2

Another blind spot is the efficiency of client-server communication. It is crucial to monitor how fast the server responds to the information given by the client, as delays can impact user experience.

To assess server responsiveness, we designed an experiment that measures the latency of the AI's responses to structured and unstructured user inputs. Timestamps are recorded for when a prompt is sent and when a response is received. We tested this under three internet conditions: optimal (fast and stable), average (typical home Wi-Fi), and constrained (limited bandwidth and increased latency using throttling tools). Additionally, we tested under different server loads: idle, moderate, and high, simulating multiple users accessing the system simultaneously. By logging response times in milliseconds, we can determine how efficiently the AI handles requests and identify delays due to network or server issues.



Condition	Avg Response Time (ms)
Optimal + Idle Load	1,800
Average + Moderate Load	3,500
Constrained + High Load	7,200

Figure 9. Figure of experiment 2

The experiment revealed a clear relationship between server load and network condition with AI response time. Under optimal conditions with an idle server, the average latency was only 1,800 ms. With moderate load and average network quality, response times nearly doubled to 3,500 ms. Under constrained conditions with high traffic, the delay more than tripled to 7,200 ms. The mean response time across conditions was 4,166 ms, the median was 3,500 ms, the lowest was 1,800 ms, and the highest was 7,200 ms. These results were expected but highlight the importance of load balancing and adaptive throttling strategies for scalability. What surprised me was how sharply response time degraded under combined strain from both poor connectivity and server overload. The single biggest factor was network bandwidth, though server load also significantly influenced performance. This suggests that implementing local caching, limiting simultaneous requests, or preloading certain AI features could greatly improve the user experience under less-than-ideal conditions.

5. RELATED WORK

A scholarly study titled Understanding Choice Independence and Error Types in Human-AI Collaboration explores how users' reliance on AI is affected by perceived performance, even in unrelated tasks [8]. The study found that users often over-trust AI systems when they perform well in complementary domains, even if they err in the user's area of expertise. This over-reliance highlights a key issue in human-AI collaboration violations of choice independence. The researchers emphasize the need for AI systems to clearly communicate their reliability and types of errors. While the study identifies this behavioral bias, it does not offer solutions for mitigating it. In contrast, our project addresses this gap by structuring user input through guided questions and programming the AI to avoid direct edits. This helps reduce over-dependence and keeps users aware of their own role in the feedback process. By reinforcing user agency and transparency, our solution improves trust calibration and maintains writing authenticity.

The study ChatGPT: Precision Answer Comparison and Evaluation Model (PACEM) by Aladdin et al. investigates the trade-off between response time and accuracy in ChatGPT [9]. Researchers found that while ChatGPT delivers highly accurate answers especially in complex subjects like literature and ethics, it does so at the cost of significantly slower response times compared to humans. For example, responses in literature took an average of 38.51 seconds, more than twice as long as human responses. Although the study effectively quantifies this issue, it does not offer practical strategies to reduce latency or improve user experience. Our project builds upon this by introducing pre-programmed questions that limit ambiguity and guide user input. This structure reduces the AI's processing burden, resulting in faster and more focused responses. Unlike the PACEM study, which passively measures delays, our solution actively mitigates them by design, ensuring both efficiency and relevance in AI-user communication within a writing support context.

A recent study titled Collaborative human-AI trust (CHAI-T): A process framework for active management of trust in human-AI collaboration introduces a dynamic, trust-centric framework designed for ongoing human-AI teaming in complex and uncertain environments [10]. Instead of treating trust as static, CHAI-T outlines how trust evolves through phases formation, calibration, adaptation by integrating team processes, performance feedback, and environmental cues. This approach resonates with our structured prompt design: by prompting guided responses and reflections, we embed trust-calibrating checkpoints throughout the interaction. CHAI-T's emphasis on active trust management informs our design, encouraging staged interventions (e.g., prompts that ask users to assess AI confidence) to maintain healthy autonomy and partnership.

6. CONCLUSIONS

While the application successfully creates a structured, supportive environment for student writers to exchange feedback and receive AI-assisted guidance, several limitations remain. First, despite the use of structured prompts to control AI behavior, the system still occasionally deviates by offering direct edits when prompted improperly [13]. Future iterations could implement stricter input validation or include real-time AI monitoring to restrict responses beyond the intended feedback-only role [14]. Second, the platform depends heavily on consistent internet connectivity and server uptime, which may limit access in low-resource environments. Implementing offline caching or lightweight fallback models could improve accessibility. Additionally, user engagement may be limited if the feedback community lacks diversity or active participation. Future enhancement might include gamification strategies or mentor-led discussion threads to maintain engagement. Finally, the current version lacks robust

personalization for writing levels or genres. A next step could be training the AI to recognize writing proficiency and genre context for tailored advice.

This application bridges the gap between peer-based feedback and AI-assisted writing support. By empowering students to learn from each other and receive structured, bias-controlled AI input, the platform enhances self-regulated learning, preserves creativity, and promotes collaborative writing habits—all while reducing overreliance on automated editing tools.

REFERENCES

[1] Zhang, Han, et al. "Online peer editing: effects of comments and edits on academic writing skills." Heliyon 8.7 (2022).

[2] Newton, Douglas P. "Quality and peer review of research: an adjudicating role for editors." Accountability in research 17.3 (2010): 130-145.

[3] Arumugam, Nalini, et al. "Prewriting discussion and academic writing." Advanced Science Letters 24.4 (2018): 2569-2572.

[4] Stasi, Roberto, and Adrian C. Newland. "ITP: a historical perspective." British journal of haematology 153.4 (2011): 437-450.

[5] Li, Zhuoyan, et al. "The value, benefits, and concerns of generative ai-powered assistance in writing." Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. 2024.

[6] McKee, Heidi A., and James E. Porter. "Ethics for AI writing: The importance of rhetorical context." Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. 2020.

[7] Hosseini, Mohammad, David B. Resnik, and Kristi Holmes. "The ethics of disclosing the use of artificial intelligence tools in writing scholarly manuscripts." Research Ethics 19.4 (2023): 449-465.

[8] Aladdin, Aso M., et al. "ChatGPT: Precision Answer Comparison and Evaluation Model." Authorea Preprints (2024).

[9] Erlei, Alexander, Abhinav Sharma, and UjwalGadiraju. "Understanding choice independence and error types in human-ai collaboration." Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. 2024.

[10] McGrath, Melanie J., et al. "Collaborative human-AI trust (CHAI-T): A process framework for active management of trust in human-AI collaboration." arXiv preprint arXiv:2404.01615 (2024).

[11] Pop, Mădălin-Dorin, and Andreas-Robert Stoia. "Improving the tourists experiences: Application of firebase and flutter technologies in mobile applications development process." 2021 International Conference Engineering Technologies and Computer Science (EnT). IEEE, 2021.

[12] Abras, Chadia, Diane Maloney-Krichmar, and Jenny Preece. "User-centered design." Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications 37.4 (2004): 445-456.

[13] Abrardi, Laura, Carlo Cambini, and Laura Rondi. "Artificial intelligence, firms and consumer behavior: A survey." Journal of Economic Surveys 36.4 (2022): 969-991.

[14] O'Reilly, Cindy A., and Andrew S. Cromarty. "" Fast" is not" real-time": Designing effective real-time AI systems." Applications of Artificial Intelligence II. Vol. 548. SPIE, 1985.

[15] Darvishi, Ali, et al. "Impact of AI assistance on student agency." Computers & Education 210 (2024): 104967.