Journal of Engineering Education Transformations,
Volume No. 37, January 2024 Special Issue, eISSN 2394-1707

A Comprehensive Pedagogical Framework for Teaching IoS App Design Course

Varsha T. Lokare[1] and Prakash M. Jadhav[2]
[1,2]Rajarambapu Institute of Technology, Sakharale, Maharashtra, INDIA
[1]varsha.lokare@ritindia.edu, [2]prakash.jadhav@ritindia.edu

***Abstract*— In modern education, equipping students with the skills and expertise to navigate the dynamic landscape of iOS app design demands a pedagogical approach that merges cutting-edge technologies with effective instructional strategies. This paper introduces a comprehensive pedagogical framework that combines the power of Visily AI, the interactive environment of Replit, and the project-based learning (PBL) approach to enhance the teaching of iOS app design. By synergizing these components, we create an immersive learning experience that caters to diverse learning styles, fosters collaboration and cultivates practical skill acquisition. Our study delves into the distinctive attributes of each tool, the theoretical underpinnings of their effectiveness, and their collective impact on student engagement and learning outcomes. Through case studies and practical implementation in an iOS app design course, we demonstrate how Visily AI aids in visual recognition tasks, Replit empowers students with a responsive coding environment, and PBL fosters real-world application and problem-solving skills. By amalgamating these elements, educators can bridge the gap between theoretical knowledge and practical proficiency. The outcomes of our approach are evidenced through improved student performance, increased motivation, and enhanced critical thinking abilities. This paper provides insights for educators seeking to enrich their iOS app design courses by harnessing the potential of Visily AI, Replit, and PBL. Ultimately, our comprehensive pedagogical framework facilitates technical competence and nurtures innovation, collaboration, and adaptability—qualities essential for success in the evolving landscape of iOS app design.**

***Keywords*— iOS App Design; Pedagogical Strategies; Project-Based Learning; Replit ;Visily AI.**

I. INTRODUCTION

The landscape of modern technology is marked by rapid evolution and innovation, compelling educators to adopt dynamic teaching methodologies that equip students with skills relevant to contemporary demands. In iOS app design education, striking a balance between theoretical knowledge and hands-on application is paramount. This paper introduces a holistic approach integrating Visily AI, Replit, and Project-Based Learning (PBL) (Kokotsaki D., 2016) to create an immersive and effective pedagogical framework for teaching iOS app design. Visily AI, a cutting-edge tool, introduces students to artificial intelligence by leveraging advanced visual recognition capabilities. Paired with Replit, an interactive coding environment, students experience a responsive platform for coding, testing, and refining their iOS app designs. In addition to these tools, the Project-Based Learning strategy involves students in practical solutions to real-world challenges, cultivating teamwork, analytical reasoning, and imaginative thinking. Integrating Visily AI, Replit, and PBL underpins the philosophy that meaningful learning emerges when theory is translated into practice. In this paper, we delve into the distinctive attributes of each component, elucidating their contributions and the synergistic effects they yield. By crafting a learning environment that resonates with diverse learning styles and instills practical skills, educators can cultivate the next generation of adept iOS app designers who are well-prepared to navigate the complexities of the digital landscape. Through case studies, student outcomes, and practical implementation, this paper showcases how the pedagogical fusion of Visily AI, Replit, and PBL (Krajcik J. S, 2006) fosters holistic learning experiences as shown in Fig.1.

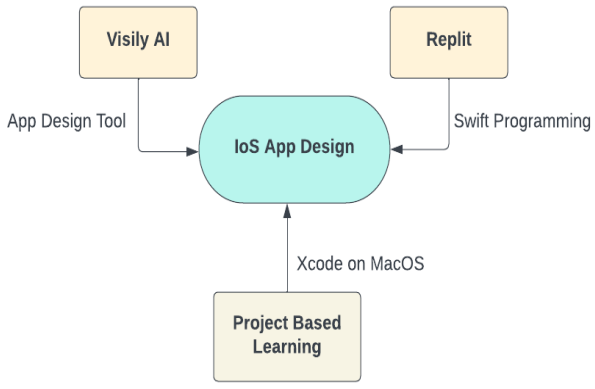

Fig.1 Activities Planned for IoS App Design Course

Ultimately, this comprehensive approach empowers students with the technical expertise needed in iOS app design and the adaptability, collaboration, and problem-solving capabilities essential for thriving in the ever-evolving technological arena. In summary, the paper is organized into distinct sections, each contributing to a comprehensive exploration of the pedagogical framework for teaching iOS app design with the integration of Visily AI, Replit, and Project-Based Learning (PBL).

Corresponding author[1]: Varsha T. Lokare, Department of Computer Science & Engineering, Rajarambapu Institute of Technology, Sakharale,
[2]Prakash M. Jadhav, Mechanical Department, Rajarambapu Institute of Technology, Sakharale,
Address: Dist: Sangli, Maharashtra, INDIA
(e-mail: varsha.lokare@ritindia.edu).

II. BACKGROUND

Integrating innovative tools and methodologies in education has been a subject of extensive research, particularly in the realm of technology-driven pedagogy. In the iOS app design education context, educators continually seek effective ways to bridge the gap between theoretical concepts and practical applications. This background study examines the existing research on integrating Visily AI, Replit, and Project-Based Learning (PBL) as a comprehensive pedagogical framework and identifies the research gap that highlights the need for a deeper exploration of their combined impact.

I. Interactive Learning using Replit: Interactive coding environments like Replit (Kovtaniuk M. S., 2023) have been explored extensively in programming education. (Cooper S.et al, 2020) delved into computer science education by investigating the effects of utilizing Integrated Development Environment (IDE) programming instead of non-IDE programming in classroom settings. Their study aimed to uncover the impact of these two distinct approaches on Computer Science (CS) majors' comprehension of programming concepts. By comparing the outcomes of both IDE-based and non-IDE-based instruction, the researchers sought to shed light on the efficacy of IDEs in enhancing programming education for students pursuing CS majors. This research contributes valuable insights to the ongoing discourse surrounding pedagogical strategies in computer science education. However, there needs to be more literature on the effectiveness of Replit in the context of iOS app design education and its integration with other tools and methodologies.

II. Visily AI and Education: Visily AI, with its visual recognition and pattern analysis capabilities, has demonstrated its potential to enhance various educational domains. The utilization of AI-driven visual recognition tools in engaging students in image analysis tasks, leading to improved learning outcomes. While this research sheds light on the potential of AI-enhanced learning, limited research directly addresses the integration of Visily AI within the context of app design education.

III. Project-Based Learning (PBL) in Technology Education: PBL has become an instructional strategy promoting active learning, critical thinking, and collaboration. Research by (R. Francese, 2015; Jumaat N. F, 2013; Nurbekova Z., 2020; Wu T. T.,2018; Al-Qora'n, 2023; Lokare, V.T. Lokare et al., 2020) explores the implementation and outcomes of incorporating Project-Based Learning (PBL) in a course focused on mobile application development. The report provides insights into the practical application of PBL methodologies in teaching students how to develop mobile applications. By sharing their experiences, challenges, and successes, the study contributes to understanding how PBL can enhance the learning process and skill acquisition in the context of mobile app development education. However, the literature needs comprehensive studies investigating the integration of PBL, Visily AI, and Replit as a cohesive framework for teaching iOS app design. While existing research provides insights into the individual potential of Visily AI, Replit, and PBL in education, a noticeable research gap exists regarding their combined integration for teaching iOS app design. Despite the advancements in technology-driven pedagogy, more studies are needed to explore the synergistic effects of these tools and methodologies on student engagement, skill acquisition, and learning outcomes within the context of iOS app design education.

This research gap underscores the need for an in-depth investigation into how the integration of Visily AI, Replit, and PBL can holistically enhance the teaching and learning of iOS app design. Such a study would contribute to the pedagogical discourse by providing educators with evidence-based insights into the effectiveness of this integrated approach and its potential to address the challenges of practical application and skill development in a rapidly evolving technological landscape.

III. INTERACTIVE LEARNING USING REPLIT

Using Replit (as shown in Fig.2) to assign Swift problem statements in a group is a convenient and effective way to engage students in collaborative programming challenges. Here is how you can use Replit to achieve this:

1. **Create a Replit Workspace:** Log in to your Replit account.
(URL: https://replit.com/)
Select the "Create" option to initiate a new workspace. Choose "Swift" as the programming language for the workspace.

2. Set Up the Problem Statement: In the Replit workspace, create a new Swift file. You can name it according to the problem statement. Write a clear and concise problem statement in the comments at the top of the file. Describe the task, input, output, and any constraints. If applicable, provide example inputs and expected outputs as comments.

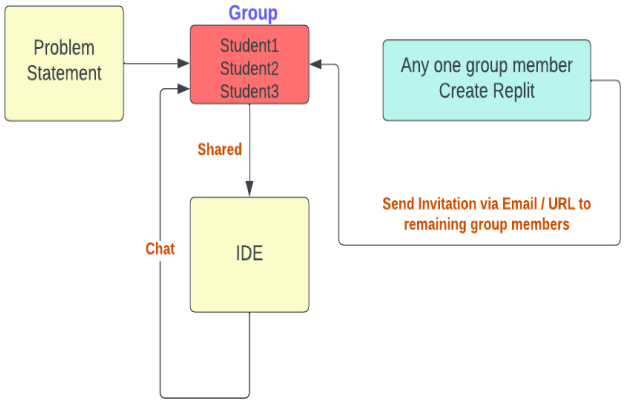

Fig.2 Replit Tool

3. **Write the Starter Code:** Below the problem statement, provide any starter code necessary for students to begin

solving the problem, like function or class definitions, variable declarations, or other necessary components.

4. Collaboration Instructions: In the comments, include instructions on how students should collaborate on the problem. You can specify whether they should work individually or in pairs/groups. If working in groups, you can suggest roles such as "coder," "debugger," and "tester" to encourage effective collaboration.

5. Share the Workspace: Once you have set up the problem statement and collaboration instructions, share the Replit workspace's URL with your students. They can access the workspace and start working on the problem statement directly in their web browsers.

6. Collaborative Problem Solving: Students can now read the problem statement, understand the requirements, and start coding within the Replit environment. They can collaborate by editing the same file simultaneously, providing comments, and discussing ideas using the integrated chat feature.

7. Code Submission and Review: After solving the problem, students can run their code within Replit to test its functionality. If you require them to submit their solutions, they can copy and paste the code or export the workspace to share the code files.

Table1: Problem based Learning in Swift using Replit

Problem Statement	**Student's Role**	**Instructors Role**	**Duration**
Implement a Fibonacci Number Calculator	- Listen to the problem statement explanation Clarify doubts about the task	- Introduce the problem statement to the class - Provide background on the Fibonacci sequence and its significance	10min
Description: Allow students to brainstorm and discuss their initial approaches individually or in groups.	- Discuss ideas within groups Outline potential strategies	- Be available for any questions or guidance students might need while brainstorming	05min
Description: Students start coding the Fibonacci number calculator function in a Replit workspace.	- Code the Fibonacci function in Replit - Collaborate within groups	- Monitor progress and help if students encounter obstacles - Provide coding tips and best practices	30min
Students test their code, identify errors, and debug as necessary to ensure correct functionality.	- Test the function with various inputs - Collaborate on debugging if issues arise	- Encourage thorough testing - Assist in debugging logic errors - Discuss common mistakes	10min
Students finalize their solutions, test them thoroughly, and prepare for submission.	- Refine the code and test it with edge cases - Prepare for submission	- Remind students of submission procedures - Address any last-minute questions	10min
Students present their solutions, explain their approaches, and discuss challenges faced.	- Present their solutions and reasoning - Ask questions and provide feedback to peers	- Facilitate a discussion where students share their code - Encourage students to explain their strategies	10min
Recap and Reflection	- Reflect on the challenges faced - Consider different approaches and lessons learned	- Recap the key concepts and techniques learned - Reflect on the importance of problem-solving in programming	05min

8. Review and Discussion: You can access each student's Replit workspace to review their solutions as the instructor. Provide feedback within Replit's comments section, addressing strengths and areas for improvement.

9. **Class Discussion:** To facilitate a class discussion, use Replit's collaboration features to bring students together in real-time to discuss different approaches and solutions.

Using Replit for assigning Swift problem statements in a group enhances collaboration, promotes hands-on learning, and allows you to provide timely feedback, making the learning experience more interactive and engaging.

By the end of this activity, students should be able to:

1. **Problem Solving:** Analyze and decompose a complex task, plan an effective approach, and implement a functional application.
2. **User Interface Design:** Design a user-friendly interface considering usability and aesthetics.
3. **Programming Skills:** Apply programming concepts to create interactive features, handle user input, and manage data.
4. **Testing and Debugging:** Test the application thoroughly to identify and resolve bugs, ensuring smooth functionality.
5. **Collaboration:** Work effectively in groups, share ideas, distribute tasks, and combine individual contributions into a cohesive project.
6. **Presentation:** Communicate the app's features, design choices, and coding challenges during the presentation.

Integrating Replit into the learning process significantly enhances problem-solving efficiency, collaboration, and the learning experience. It eliminates environment setup barriers, fosters real-time collaboration, offers immediate feedback, and promotes active learning. Students can focus more on coding, debugging, and problem-solving, leading to improved understanding and skill development. Additionally, Replit's accessibility and consistent environment foster engagement and consistent practice, contributing to enhanced learning outcomes.

IV. Project Based Learning with Visily AI Tool

The iOS App Design process has been greatly streamlined and enhanced with the introduction of Visily AI, a cutting-edge design tool. Visily AI combines the power of artificial intelligence and design principles to facilitate the creation of visually appealing and user-friendly iOS applications. Visily AI offers rapid prototyping capabilities, allowing designers to visualize app layouts and interactions quickly. The AI tool analyzes user preferences and design trends to suggest optimal layouts, improving user engagement. Visily AI generates harmonious color palettes based on the app's theme, reducing the need for manual color selection. The tool assists in creating custom icons and graphics, aligning them with Apple's design guidelines for iOS. Visily AI ensures that app layouts adapt seamlessly to different device sizes and orientations, enhancing user experience. The AI tool provides usability feedback, identifying potential user flow issues and suggesting improvements. Visily AI accelerates the app design process by automating various design tasks, saving time and resources. Visily AI revolutionizes iOS app design by leveraging artificial intelligence to expedite and enhance the design process. Its features, like intelligent layout suggestions, automated asset generation, and interactive prototyping, empower designers to efficiently create visually stunning and user-centric iOS applications.

Here are the steps to use Visily AI for your app design:

1. Sign Up and Log In: Visit the Visily AI website and create an account.
(URL: https://app.visily.ai/login?type=team&)

2. Create a New Project: Click "New Project" or a similar option to start a new design project.

3. Define Project Details: Name your project and specify the platform (iOS, in this case). Optionally, please briefly describe your app and its target audience.

4. Choose a Design Style: Select a design style or theme that aligns with your app's purpose and audience. This choice will influence the AI's design recommendations.

5. Initial Input: Provide initial information about your app, such as its primary features, key screens, and any branding elements you have.

6. Wireframing: Begin with creating wireframes for the key screens of your app. Use Visily AI's tools to quickly sketch the layout, content placement, and navigation flow.

7. Layout Suggestions: Let the AI analyze your wireframes and provide layout suggestions. Review the suggestions and decide which ones align with your vision.

8. Color Palette and Styling: Select or customize a color palette for your app. Use AI-generated color schemes that complement your chosen design style.

9. UI Component Generation: Create UI components like buttons, icons, and navigation elements. You can either customize existing components or use AI-generated ones.

10. Interactive Prototyping: Start building interactive prototypes using the designed components. Add screen transitions, animations, and interactive elements to showcase user flow.

11. Usability Testing Simulation: Simulate usability testing scenarios to see how users might interact with your prototype. Identify potential issues and make design adjustments as needed.

12. Collaboration and Feedback: Invite team members to collaborate on the project. Share the prototype and gather feedback on design, interactions, and usability.

13. Refinement and Iteration: Based on feedback, iterate on your design, making necessary changes and improvements.

14. Export Design Assets: Export the design assets once your design is finalized. This might include icons, graphics, images, and UI components.

15. Integration with Development: Provide the exported assets to your development team for integration into the app development process.

16. Continuous Improvement: Use Visily AI for design updates and improvements as your app evolves.

Remember that the steps might vary slightly based on the specific features and capabilities of Visily AI. Always refer to the tool's official documentation and guides for accurate instructions.

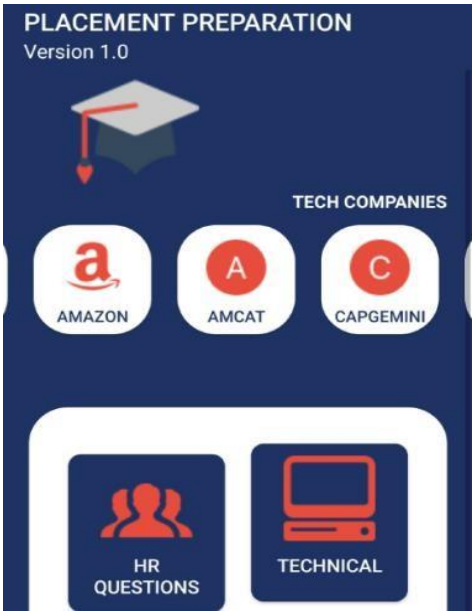

Fig.3 Sample designs created by Visily AI

Table 1. Project Based Learning Activity Plan

Activity Stage	Objective	Description	Assessment
App Idea Selection	Creativity, Problem Identification	Students brainstorm and propose ideas for iOS apps that address a specific problem or cater to a particular need. They research existing apps, identify gaps in the market, and select an app idea that aligns with their interests and market demand.	Originality of App Ideas, Relevance to Identified Problem or Audience Need, Rationale behind Idea Selection.
Market Survey	Market Awareness, User Analysis	Students conduct a market survey to understand the target audience's preferences, pain points, and expectations from similar apps. They analyse competitors' offerings, reviews, and ratings to identify opportunities for differentiation and improvement.	Depth of Market Research, Identification of User Preferences, Competitor Analysis, Insights for App Differentiation.
User Requirements	User-Centric Design, Requirement Gathering	In teams, students define and document detailed user requirements by considering the insights from the market survey. They outline essential features, functionalities, and user experience elements that the app should incorporate to meet user expectations.	Clarity and Completeness of User Requirement Documentation, Alignment with User Insights.
iOS Design with visily AI	UI/UX Design, Collaboration	Using visily AI , students design the user interface (UI) and user experience (UX) of their iOS app. They create wireframes and interactive prototypes, ensuring an intuitive and visually appealing design that caters to the identified user requirements and aligns with iOS design guidelines	UI/UX Creativity, App Navigation Logic, Incorporation of User Requirements, Adherence to Design Principles.

Implementation on Xcode	Programming Skills, Technical Proficiency	Students begin developing the app using Xcode, coding the functionalities outlined in the user requirements and UI design. They ensure proper structuring of the code, integration of user interactions, and management of app data.	Functionality and Logic of App Features, Code Structure, App Responsiveness, Handling of User Interactions.
Testing	Quality Assurance, Problem-Solving	Students rigorously test the app's functionalities to identify bugs, glitches, and inconsistencies. They gather feedback from users and peers to uncover areas of improvement. They iteratively address the identified issues, refine the design, and optimize the user experience based on testing outcomes.	Identification and Resolution of Bugs, Usability Testing Results, Incorporation of Feedback, Overall App Perform

Outcomes: As mentioned below, total 12 iOS apps are designed to cater to various learning, informational, and productivity needs, allowing users to acquire new skills, enhance their knowledge, and improve efficiency in various domains.

1. C Programming App:

Outcome: Learn the fundamentals of C programming, practice coding exercises, and develop problem-solving skills in the language.

2. Prodigy Maths Kids Learning App:

Outcome: Enhance mathematical skills through engaging games, puzzles, and interactive challenges tailored for kids.

3. Placement Preparation App:

Outcome: Prepare for job placements with mock tests, interview tips, and resources to improve technical and soft skills.

4. Event Management App:

Outcome: Manage events efficiently by organizing schedules, sending invitations, tracking attendees, and coordinating activities.

5. Restaurant Menu App:

Outcome: Explore restaurant menus, place orders, and facilitate seamless dining experiences for customers.

6. Python Learning App:

Outcome: Acquire proficiency in Python programming through interactive lessons, coding exercises, and real-world examples.

7. Career Guidance App:

Outcome: Receive personalized career advice, explore various career paths, and access resources for skill development.

8. Around U App:

Outcome: Discover nearby points of interest, get directions, and explore local businesses with geolocation services.

9. Java Learning App:

Outcome: Gain a comprehensive understanding of Java programming, learn concepts, and practice coding challenges.

10. Machine Learning App:

Outcome: Delve into machine learning concepts, algorithms, and applications through tutorials, projects, and interactive content.

11. Expense Tracker App:

Outcome: Monitor and manage personal expenses, track budgets, and visualize spending patterns for financial management.

12. HTML Tags & Attributes App:

Outcome: Learn HTML markup language by studying different tags and attributes, fostering web development skills.

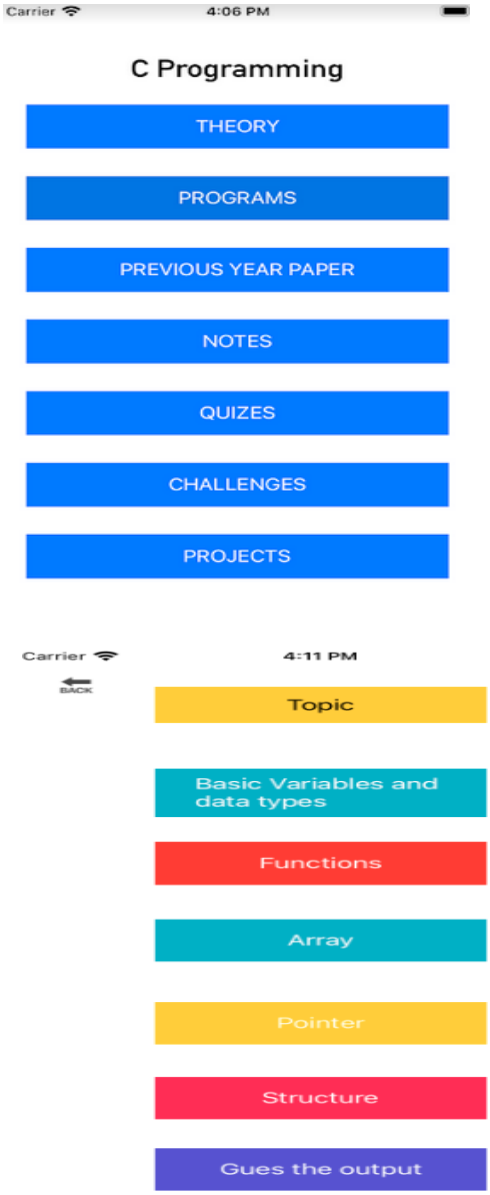

Fig.4 Screenshots of sample IOS Apps

V. FEEDBACK ANALYSIS

Here is an analysis of the feedback responses based on the questions provided:

1. Did working on real-world projects through PBL help you develop practical skills and problem-solving abilities?

Yes (67 responses): Most students found that working on real-world projects through PBL positively contributed to their practical skills and problem- solving abilities.

This indicates that the hands-on nature of PBL effectively enhanced their ability to apply theoretical knowledge to real-life scenarios.

No (2 responses): A very small number of students indicated that they did not perceive a significant improvement in practical skills and problem-solving abilities through PBL.

Further investigation might be needed to understand the specific challenges or areas where improvement is desired.

May Be (6 responses): Some students expressed uncertainty about how much PBL contributed to their practical skills and problem-solving abilities. This could suggest that these students might need additional clarification or examples of how PBL aligns with skill development.

2. How did Visily AI enhance your app design process compared to traditional design tools?

Yes (70 responses): Most students reported that Visily AI noticeably improved their app design process compared to traditional design tools. This indicates that Visily AI's features, such as layout suggestions and color palettes, effectively streamline and enhance the design workflow.

No (0 responses): None of the students indicated that Visily AI did not enhance their app design process. This suggests that the tool was generally well-received and lived up to the expectations of the students.

May Be (5 responses): A small number of students expressed uncertainty about the extent of Visily AI's impact on their app design process. This might imply that these students may have had mixed experiences or need further guidance on utilizing the tool effectively.

3. Did Replit's online collaboration features enhance your group projects and coding collaboration?

Yes (74 responses): A large majority of students found Replit's online collaboration features to be beneficial for their group projects and coding collaboration. This indicates that the collaborative environment provided by Replit effectively supported teamwork and coding together.

No (0 responses): No students reported that Replit's collaboration features did not enhance their group projects or coding collaboration. This suggests that the collaborative aspects of Replit were universally well-received among the students.

May Be (1 response): A single response indicated uncertainty about the extent of Replit's impact on group projects and collaboration. This could signify that this student might have encountered specific challenges or need further clarification on utilizing Replit's collaboration features. This feedback analysis provides insights into students' perceptions of the effectiveness of Project- Based Learning (PBL), Visily AI, and Replit. Most students have positive experiences with these methods and tools, but a few students also expressed uncertainty or encountered challenges. Further exploration and communication with these students could uncover specific areas for improvement and refinement. Overall, this feedback analysis emphasizes the positive impact of these tools and methods on students' learning journey. However, it also underscores the importance of continuous improvement and individualized support to cater to diverse learning preferences and needs. The insights gained from student feedback offer valuable guidance for refining these approaches, fostering a more engaging, efficient, and effective educational experience. By incorporating these insights, educators can continue to create environments that empower students to excel and flourish in their learning endeavors.

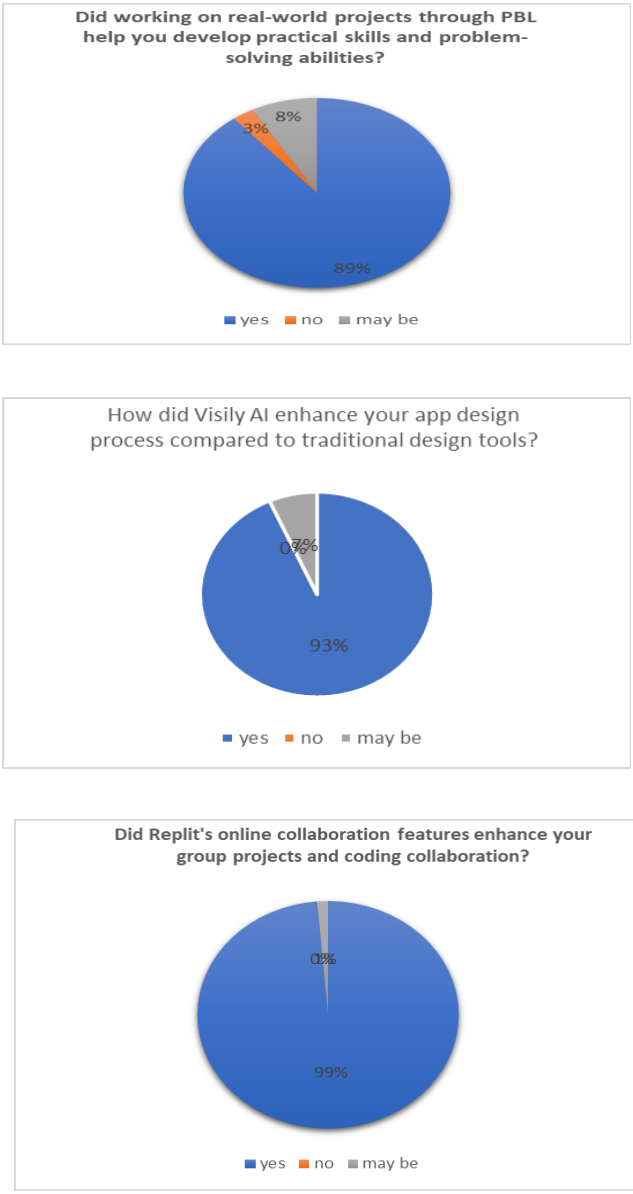

Fig.5 Feedback Analysis

VI Performance Metric Analysis

I. *Parameter: Score in Examination*

The average marks for students in the PBL group were significantly higher (72.5) compared to those in the Non-PBL(Prince K. J, 2005; Couto L. B, 2015; Valdez J. E, 2019; Wun Y. T, 2007) group (48.74). This suggests that the PBL approach may have a positive impact on student performance (as shown in Fig.6)

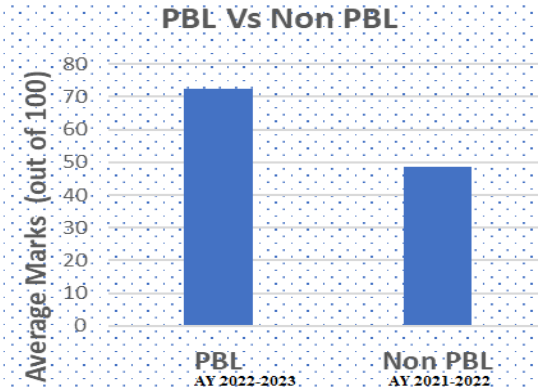

Fig.6 PBL Vs Non PBL

I. *Parameter: Attendance*

The Pearson correlation method has been applied to measure the relation between students' attendance in lab and lecture sessions and their exam scores. PBL and attendance (r = 0.87) and Non- PBL and attendance (r = 0.66). Here's an analysis of these correlation coefficients: **PBL and Attendance (r = 0.87):** A correlation coefficient 0.87 indicates a strong positive linear relationship between PBL and attendance. The positive sign of the correlation coefficient (0.87) indicates that when PBL performance goes up, attendance also tends to go up. **Non-PBL and Attendance (r = 0.66):** A correlation coefficient 0.66 indicates a moderately strong positive linear relationship between Non-PBL performance and attendance. Similar to the PBL group, this suggests that as Non-PBL performance increases, attendance tends to increase, or vice versa. However, the correlation in this case is slightly weaker than in the PBL group (0.66 vs. 0.87). In both cases, a positive correlation suggests that students who perform well (whether in PBL or non-PBL) tend to have better attendance records, which is an expected and positive association. The difference in the correlation strength implies that PBL has a stronger influence on attendance compared to the Non-PBL approach, but other factors may also be at play.

	PBL	*Attendance*
PBL	1	
Attendance	0.87	1

	Non PBL	*Attendance*
Non PBL	1	
Attendance	0.66	1

CONCLUSION

The demand for well-designed and user-friendly iOS applications is incessant. Companies prioritize mobile app development, creating a consistent need for skilled iOS designers and developers. In the ever evolving realm of modern education, proficiency in iOS app design is pivotal. This paper unveils a groundbreaking pedagogical framework, seamlessly blending cutting edge technologies like Visily AI, the interactive landscape of Replit, and the project-based learning (PBL) paradigm. This synthesis forms a transformative path for teaching iOS app design, catering to diverse learning styles, fostering collaboration, and cultivating essential practical skills. Our meticulously crafted framework amalgamates these components to create an immersive learning journey. By exploring each tool's intricacies, elucidating their theoretical underpinnings, and assessing their collective impact on engagement and learning outcomes, we usher in a new pedagogical era. Real-world case studies and practical implementations within an iOS app design course highlight the transformative potential of Visily AI's visual recognition, Replit's coding environment, and PBL's problem-solving capabilities. This synthesis bridges the gap between theory and proficiency, facilitating holistic student growth. The outcomes are tangible enhanced student performance, increased motivation, and sharper critical thinking. This paper isn't just theoretical; it's a guide for educators seeking to enrich their iOS app design courses. By leveraging Visily AI, Replit, and PBL, educators can equip students with essential skills for success in iOS app design. The findings underscore the PBL approach's efficacy in attendance and performance compared to non-PBL approaches.

In summary, our pedagogical framework champions technical competence, innovation, collaboration, and adaptability crucial qualities for triumphing in the constantly shifting landscape of iOS app design. As education and technology intertwine, this paper lights the path toward holistic learning experiences, preparing students for today's challenges and the limitless opportunities of tomorrow.

References

Kovtaniuk, M. S. (2023). Online compiler «Replit» usage during the study of the Programming discipline. *Publishing House "Baltija Publishing*.

Cooper, S., Clinkscale, B., Williams, B., & Lewis, M. (2020, February). Exploring the impact of exposing CS majors to programming concepts using IDE programming vs. non-IDE programming in the classroom. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 1422-1422).

Francese, R., Gravino, C., Risi, M., Scanniello, G., & Tortora,G. (2015). Using Project-Based-Learning in a mobile application development course—An experience report. *Journal of Visual Languages & Computing*, *31*, 196-205.

Jumaat, N. F., & Tasir, Z. (2013). Integrating project-based learning environment into the design and development of mobile apps for learning 2D-animation. *Procedia-Social and Behavioral Sciences*, *103*, 526-533.

Nurbekova, Z., Grinshkun, V., Aimicheva, G., Nurbekov, B., & Tuenbaeva, K. (2020). Project-based learning approach for teaching mobile application development using visualization technology. *International Journal of Emerging Technologies in Learning (iJET)*, *15*(8), 130-143.

Wu, T. T., Huang, Y. M., Su, C. Y., Chang, L., & Lu, Y. C.(2018). Application and analysis of a mobile e-book system based on project-based learning in community health nursing practice courses. *Journal of Educational Technology & Society*, *21*(4), 143-156.

Al-Qora'n, L. F., Jawarneh, A., & Nganji, J. T. (2023). Toward Creating Software Architects Using Mobile Project-Based Learning Model (Mobile-PBL) for Teaching Software Architecture. *Multimodal Technologies and Interaction*, *7*(3), 31.

Lokare, V. T., Jhetam, M. I., & Netak, A. K. D. L. (2020). A Blended Approach for Teaching Laboratory Course: Internet Technology. Journal of Engineering Education Transformations, 34(1), 41-51.

Krajcik, J. S., & Blumenfeld, P. C. (2006). Project-based learning (pp. 317-34). na. Kokotsaki, D., Menzies, V., & Wiggins, A. (2016). Project- based learning: A review of the literature. Improving schools, 19(3), 267-277

Prince, K. J., Van Eijs, P. W., Boshuizen, H. P., Van Der Vleuten, C. P., & Scherpbier, A. J. (2005). General competencies of problem-based learning (PBL) and non-PBL graduates. Medical education, 39(4), 394-401.

Couto, L. B., Bestetti, R. B., Restini, C. B., Faria-Jr, M., & Romão, G. S. (2015). Brazilian medical students' perceptions of expert versus non-expert facilitators in a (non) problem-based learning environment. Medical Education Online, 20(1), 26893.

Valdez, J. E., & Bungihan, M. E. (2019). Problem-based learning approach enhances the problem solving skills in chemistry of high school students. JOTSE, 9(3), 282-294.

Wun, Y. T., Tse, E. Y., Lam, T. P., & Lam, C. L. (2007). PBL curriculum improves medical students' participation in small- group tutorials. Medical teacher, 29(6), e198-e203

Author1: Prof. Varsha T. Lokare is currently pursuing a Ph.D. in the Brain-Computer Interface domain. She received the B.Tech degree in Computer Engineering from the Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, and M.Tech degree in Computer Science and Engineering from Rajarambapu Institute of Technology, Sakharale and currently pursuing PhD from the Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad. Her research interest includes Brain Computer Interface, AI, Opportunistic Routing, Multi-criteria decision- making, Algorithms, and Networking. She has published many papers on BCI domain, Opportunistic Routing, Multi-criteria decision making, and Pedagogy. She has 17 years of teaching experience and is currently working as Assistant Professor at Rajarambapu Institute of Technology,Sakharale.

Author2 : Prof. Prakash Jadhav received the B.E. and M.E. degree in Mechanical Engineering from Shivaji University. His research interest includes Neural Networks and vibrations. He has 20 years of teaching experience and two years of industrial experience in software. He has published many papers on Vibration and Design, Multi-criteria decision making, and Pedagogy. He is currently working as Assistant Professor at Rajarambapu Institute of Technology,Sakharale.