JOURNAL OF EDUCATIONAL RESEARCH & PRACTICE

Journal of Educational Research & Practice
2022, Volume 12, Issue 1, Pages 179–189
DOI: 10.5590/JERAP.2022.12.1.13

Original Research

WALDEN UNIVERSITY

Improving Computer Programming Competency for First Semester Computer Science Students Through Immersive Project-Based Learning

Ubaidah Ubaidah, M. Pd.
Binus University, Jakarta, Indonesia
https://orcid.org/0000-0002-6873-0160

Minaldi Loeis, M.Sc., M.M.
Binus University, Jakarta, Indonesia
https://orcid.org/0000-0002-1138-624X

Contact: *sanasini.ubai@gmail.com*

Abstract

The objective of this research is to describe the implementation of project-based learning (PJBL) in improving computer programming competency in a higher education setting. The method applied in this study is action research with a one-cycle framework with four phases of development: a) planning, b) action, c) observing, and d) reflection phases. This research tries to answer two questions: how to implement PJBL in a programming course to improve programming competency and how to ensure students' satisfaction in the learning process. The PJBL applied in this research consists of seven steps, including a) a challenging problem or question; b) sustained inquiry; c) authenticity; d) student voice & choice; e) reflection; f) critique and revision; and g) public product. We found that the PJBL implementation for the course, Introduction to Programming, supports the attainment of programming competence. We found that PJBL helps students retain programming knowledge and transfer the programming skills into real-life projects. The research also found that with the implementation of PJBL in Introduction to Programming, students' satisfaction with the learning experience increases. The result shows that PJBL is not only effective in improving computer programming competency, but also in increasing the student's engagement and passion for lifelong learning.

Keywords: *project-based learning, programming competency; higher education*
Date Submitted: June 20, 2022 | **Date Published:** October 3, 2022

Recommended Citation

Ubaidah, U., & Loeis, M. (2022). Improving computer programming competency for first semester computer science students through immersive project-based learning. *Journal of Educational Research and Practice, 12*, 179–189. https://doi.org/10.5590/JERAP.2022.12.1.13

Introduction

Indonesia has a population of over 267 million people, ranking fourth in the world in terms of population, and human resources are one of its potential resources for future economic growth for the country. However, the

Indonesian human resources competitiveness ranking in the IMD World Talent Ranking 2019 shows Indonesia ranked at 32, indicating that the Indonesian education system still has many challenges to overcome. This fact also shows that Indonesian higher education systems should create breakthroughs to ensure its graduates are ready to compete in the world's knowledge economy and able to contribute to society as problem solvers.

Since many Indonesians consider university education the final stage of formal learning before entering the job market, it is important for universities to conduct teaching and learning activities that not only expose students to knowledge comprehension but also to the skills to creatively solve real-world problems. One effective learning strategy to answer this need is project-based learning (PJBL). PJBL is a learning strategy that focuses on integrating learning into a real-life problem with the hope that it will trigger students to apply their knowledge and skills to a feasible solution (Bender, 2012; Isomottonen et al., 2019; Kokotsaki et al., 2016). Through PJBL, students are exposed to meaningful experiences that prompt them to actively and creatively integrate conceptual knowledge into applied solutions for solving problems.

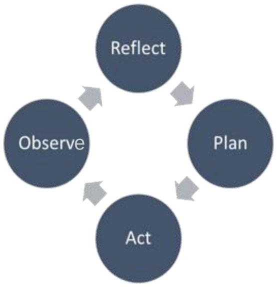
Some researchers have shown that PJBL could be an impactful learning strategy. PJBL is a highly effective learning strategy that nurtures students to become the competent, caring, and ethical healthcare professionals of the 21st century (Bender, 2012; Lee et al., 2016). Montero and Warwick (2018) conducted research on PJBL for science, technology, engineering, arts, and mathematics (STEAM) learning, and the results show that PJBL can contribute to the improvement of learning results, as well as improve the self-regulated learning skills of students. (Montero & Warwick, 2018). Bender also conducted research in PJBL, showing that PJBL fosters 21st-century skills and innovative thinking. He also mentioned that PJBL is a model of instruction for the 21st century and the basis for lifelong, anytime, anywhere learning (Bender, 2012).

In enriching the research on PJBL, this study aimed to describe the implementation of PJBL in improving computer programming competency in a higher education setting. The main components of PJBL in this research are (a) a real-world problem brought into the classroom designed to stimulate the active engagement of students through problem-solving, (b) a student-centered learning environment where lecturers have a role as facilitators to enable active and collaborative learning, and c) a focus on learning outcomes in developing skills, motivations, and passion for lifelong learning. This research will answer two research questions: 1) how to implement PJBL in a programming course to improve programming competency, and 2) how to ensure students' satisfaction in the PJBL learning process.

Research Method

In this study, we used the Action Research method to apply PJBL in a higher education setting to improve computer programming competency. The setting for this research is an international class in a programming course at Binus University. The action research process involves the reflecting, planning, acting, and observing phases (Brendel & Cornett-Murtada, 2019). As shown in Figure 1. The "reflecting" phase is also referred to as identifying and limiting the topic by reviewing what is already known about the problem or focus area and learning more about the problem. In the "planning" phase the research questions and methods are explicated, and a plan is developed for acting. In the "action" phase, researchers implement the plan or change practice and collect data. As in the "observation" phase, researchers synthesize and analyze the data to find out whether the problems have been solved or need more improvement. The cycle is repeated until the learning needs have been resolved (Brendel & Cornett-Murtada, 2019).

Figure 1. *Action Research Process*



Reflecting Phase

Since 2016, Binus University has implemented a “block mode” approach system where the curriculum includes critical courses designed as blocks of subject matter that are taught intensively and therefore completed more quickly compared to the regular courses, which are designed to have face-to-face sessions every week throughout the semester. Like any regularly running courses, the block courses are based on the expected competencies to be delivered; however, due to the intense learning, students can focus on comprehending one competency at a time. A typical block-mode course is designed to run/be conducted on 3 consecutive days weekly and completed within 3 months. Therefore, we believe that in the block mode setting, the course design needs improvement to help students achieve programming competency in allocated times and to ensure the students’ satisfaction with the learning process. Based on those needs, we conducted literature reviews that led to the application of PJBL to Introduction to Programming courses.

Planning Phase

In designing the course, we created the course outline in such a way that it meets the needs and/or requirements of the expected student outcomes, university policies, and the time allocated for the subject (semester schedule). The Introduction to Programming course was designed as the first programming course for first-year students majoring in computer science. One expected student outcome for the course is for students to be able to design a software application as a solution. Students should develop solutions that are based on a formal structured approach and problem analysis in computing. In supporting the expected student outcomes within the allocated time, the course is designed as an intensive block mode course in combination with another course that, in the past, focused on problem analysis. The course uses the Python programming language, which is easier to learn compared to other programming languages due to its simplicity in implementing data structures. Students can overcome the difficulty of learning a programming language more quickly and, therefore, the course offers a quicker learning path to reaching a competent level for writing programs and solving problems through developing software applications.

To maintain the constructive alignment in course design, the learning outcomes were designed based on expected student outcomes set by the curriculum. After completing the Introduction to Programming course, students will be able to: a) apply principles of Python programming in designing and developing a program; b) demonstrate knowledge and skills pertaining to the fundamental building blocks of object-oriented programming; c) use data to generate interactive visualizations.

To support the expected course learning outcomes, the Introduction to Programming Course has a load of 6 credits, with a total of 26 sessions in 7 weeks. Generally, the course was designed into two main sessions, the first session is a scaffolding session that focuses on introducing new concepts and approaches that can be applied by students to their projects. The second session focuses on how students engage in their project;

based on real problems and is set in such a way that the students must apply the concepts and approaches they have learned in the previous session.

Another output in the Course Design phase is the assessment plan. The most important thing in planning assessment is how to ensure the alignment between assessment and intended course learning outcomes. Therefore, we mapped the intended course learning outcomes to assessment. Table 1 shows the mapping of the assessment method to course learning outcomes:

Table 1. *Mapping of Assessment Method to Course Learning Outcomes*

Assessment	Weight	CLO 1	CLO 2	CLO 3
Final project	60%	✓		✓
Final exam	40%	✓	✓	

Based on the mapping, we decided to conduct a final project and final exam as the main assessment component to be evaluated on ensuring the achievement of course learning outcomes. The final project requires students to develop software applications based on real-life problems using Python programming. Students present the development process of software application and how it works. As the preparation for conducting the assessment, researchers also write assessment rubrics.

Another factor to be prepared for implementing the course design is the lecturer's skills in implementing PJBL. It was important to carefully invest in the necessary support in making PJBL a success. Investments include lecturer training and learning resources. During this preparation phase, a heavy emphasis was put on preparing lecturers so that they became effective PJBL facilitators as well as ensuring preparedness of all media and instruments that support the new teaching and learning processes for the new course outline.

Preparing lecturers to become PJBL facilitators was deemed important since, in PJBL, the role of a facilitator requires the ability in asking stimulating questions to trigger critical thinking, help students acquire knowledge, create an environment for students to develop and implement social skills, and assess students' performance (Bender, 2012; Efstratia, 2014; Thomas, 2000). We conduct a faculty development program (FDP) for PJBL by providing a training series on conducting PJBL, developing and using case studies in the classroom, giving meaningful feedback, and authentic assessment.

We also conducted training in preparing supported media and instruments to ensure that teaching and learning activities aligned with the designed course to support learning outcomes achievement. We also considered it important to have all the necessary media and instruments ready prior to the implementation of teaching and learning activities to ensure quality. Specific to the Introduction to Programming course, we developed several case studies, presentation slides, tutorial videos, and assessment instruments.

Action Phase

The Introduction to Programming course was taught by a team of lecturers and lab assistants. The teaching of concepts and problem-solving exercises were carried out by lecturers as scaffolding, while lab assistants assisted students in practicing and providing tutorials on students' programming skills. Lecturers also conducted assessments and evaluations.

This research adopted the Gold Standard Project-Based Learning by PBLWorks (PBLWorks, 2022). The seven essential project design elements are:

1. a challenging problem or question
2. sustained inquiry

3. authenticity
4. student voice and choice
5. reflection
6. critique and revision
7. public product

The final project for the course was a project on application development. Students had a briefing on the final project from the first session. The project was a group project, and the development process was carried out from the early session of the course based on the provided problem. Students had the opportunity to directly practice the concepts that were obtained in their project. Both lecturers and lab assistants were ready to facilitate the students in finishing their projects. Students went through seven stages in the PJBL to complete their projects as follows:

A challenging problem or question
In this phase, students were required to look for problems or questions in the community that can be solved through software development. The lecturer equipped students with some briefings about programming concepts, criteria, and principles to help students define problems or questions to be solved through software. After clarification of the problem or question, each group then discussed and reached an agreement on the root problem that can be solved by developing a software application. The definition of the problem was the main goal during this phase.

Lecturers ensured whether the problems or questions found by students were worthy of being the basis of project development. Lecturers placed themselves as facilitators ready with various possible problem definitions without trying to drive the problem into the individual perspective of the lecturer.

Sustained inquiry
Based on the defined problem, each group then started brainstorming. The purpose of the group brainstorming discussion was to let everyone express their ideas about the problems. The ground rules for this phase were to let everyone freely throw out their ideas without immediate discussion and comment. Each group then compiled ideas of the underlying circumstances of the problem and possible implications arising from the problem.

Based on the underlying circumstances of the problem, each group started to connect each idea to the structured problem and alternative solutions and present it to the class. In this phase, students also listed their prior knowledge that relates to the problems. This prior knowledge may be based on information acquired in earlier education, facts, and insights obtained by reading different articles or in another way. The facilitator also urged students to realize the knowledge and equipment that students must accomplish for the project. In this condition, since students' prior knowledge was limited, each group started to question the feasibility of the project. This is the time when the facilitator encouraged students to define the gap as learning goals and put them in charge of learning. Facilitators provided answers to the questions evoked in the prior phase and offered students the possibility to acquire a more profound knowledge of computer programming that relates to the project. Students explored relevant sources of knowledge and then put the new information together to equip them to accomplish their projects.

Authenticity
As the structure of the problem was already set, each group was then ready to set the objectives of the project. In this phase, the facilitator ensured the project objectives were focused, achievable, comprehensive, appropriate, and feasible.

Student voice and choice

Based on the set project's objective, students proposed their plan for executing the project. The planning included milestones, targets, approaches, and related resources. The lecturer allowed the students to express their own voices. When the lecturer found some impossible timelines presented by students, the lecturer's role was only on briefing the criteria and standards to make students realize the feasibility of their project. The lecturer gave a space for students to set their own strategies and express their voices.

Reflection

As the development phase was running, students were aware of the learning process that they experienced through project completion. Students had to present their project development progress and answer some reflection questions to make them reflect on their learning process. Through this presentation, the lecturer monitored the standard and quality of students' work and gave some advice for students' obstacles, if needed.

Critique and revision

In addition to helping students reflect on their learning process, the project development presentations were also intended to assist students in getting input and criticism of the software they developed. Based on these inputs and criticisms, given by lecturers and fellow students, each group could improve its software

Public product

After going through the review and revision process, the product was delivered in the final version. Students presented the results of their project including the implications for solving problems or questions that underlie the development of the project. Each group presented their project result. The facilitator assessed the project. As for the closing of the learning process, the class discussed whether they acquired a more proficient, accurate, detailed explanation and understanding of what was going on behind the problem. As for the final exam, the students were completing the final exam to ensure knowledge comprehension.

Result

Observation Phase

In the observation phase, we reevaluated whether the problems that were analyzed in the reflection phase had been solved. In the reflection phase, the learning done by students needs to cover: a) achieving the programming competency in allocated times and b) ensuring the satisfaction of students with the learning process. To identify those needs that have been met, an evaluation of the learning outcomes and a satisfaction survey are conducted in this phase.

In evaluating the learning outcomes achievement, an examination of students' assessment results of the course learning outcomes (CLO) is done. Rubrics used for assessment are calibrated into four achievement levels: Excellent, Good, Average, and Poor. The criteria of the rubrics can be seen as follows:

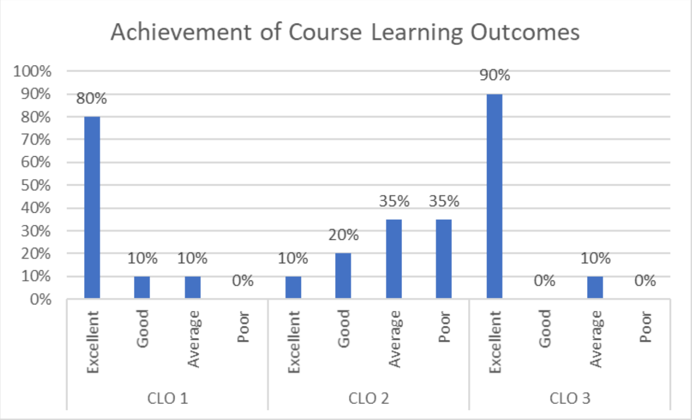
1. **CLO 1:** Apply principles of Python programming in designing and developing a program
   Criteria 1.1: Program alignment with the intended objectives
   Criteria 1.2: efficiency of code

2. **CLO 2:** demonstrate knowledge and skills pertaining to the fundamental building blocks of object-oriented programming
   Criteria 2.1: knowledge comprehension

3. **CLO 3:** use data to generate interactive visualizations
   Criteria 3.1: interactivity of visualization

The student achievements of each CLO are graphically represented in Figure 2. The graph shows the achievements of various CLOs in the percentage of students in the class. How the CLOs are assessed is given in Table 1. CLO 1 is assessed through a final project and final exam, CLO 2 is assessed through a final exam, and CLO 3 is assessed through the final project.

Figure 2. *The Achievement of Course Learning Outcomes*



The data and its evaluation show that the PJBL implementation for the Introduction to Programming with the intention to deliver programming competency, which is to design and develop a program (CLO 1), for first-semester students using the Python programming language was achieved successfully. The result shows that 80% of students achieved “excellent” performance. Similar results for the competence of “using data to generate interactive visualizations” which is represented by CLO 3; results show that 90% of the students achieved “excellent” performance. This shows that PJBL can support the achievement of competence in cognitive higher-order thinking based on Bloom’s taxonomy.

However, students’ competency attainment of “demonstrating knowledge and skills pertaining to the fundamental building blocks of object-oriented programming” (CLO 2) resulted unfavorably; 35% of students achieved Poor results, 35% achieved Average results, 20% achieved Good results, and only 10% achieved Excellence. This shows that from the PJBL experimentation, the course needs to be improved, especially regarding the programming fundamentals of object orientation. Furthermore, the process of retention of knowledge needs to be improved so that meaningful learning can be attained.

After evaluating the students’ achievement data, we conclude that the PJBL implementation for the course Introduction to Programming supports the attainment of programming competence. Some aspects of teaching and learning, such as the process of knowledge retention of fundamental principles of programming, need to be improved to ensure students’ mastery of knowledge.

In identifying students’ satisfaction with the learning process, we conducted a satisfaction survey, which included aspects of the course being surveyed: a) course management, b) teaching and learning method, and c) overall satisfaction. Course management is further broken down to questions in areas of the course management processes, availability of course learning materials, and appropriateness of the course towards competencies and knowledge to be attained. As for the teaching and learning aspect, we split questions into areas of delivery of course materials by the instructor, learning strategy, the effectiveness of feedback and assessment, and the instructor's commitment to facilitating students' learning needs.

The student's satisfaction survey uses a Likert scale of 1 to 6 where "1" corresponds to Very Unsatisfied, "2" for Unsatisfied, "3" for Fairly unsatisfied, "4" for Fairly satisfied, "5" for Satisfied, and "6" for Very satisfied for rating students' perception. The result of the survey is then mapped and interpreted to six levels of satisfaction described in Table 2.

Table 2. *Range of Students' Satisfaction Score*

Values	Description
5.23–6	Very satisfied
4.39–5.22	Satisfied
3.55–4.38	Fairly satisfied
2.71–3.54	Fairly unsatisfied
1.84–2.67	Unsatisfied
1–1.83	Very Unsatisfied

From averaging the data of the "learning" component of the student's satisfaction survey, which resulted in an overall average score of 4.89, it shows that most students were satisfied with the learning experience from the Introduction to Programming course, which implemented PJBL. From the "course management" component, which resulted in an overall average score of 4.82, we concluded that students were satisfied with the management, content availability, and alignment with the intended learning outcomes. As for the teaching and learning method, the average total score was 4.96, implying that the students were satisfied with the design and implementation of the PJBL in the Introduction to Programming course.

Finally, from the "overall satisfaction" component of the survey, the survey resulted in an average of 4.90, which also shows that generally, students are satisfied with taking and learning from the course. This result is higher compared to last year's course which did not use PJBL. The following comparison in Table 3 shows some contrasts before and after implementing PJBL.

Table 3. *Comparison of Students' Satisfaction Before and After Implementation of PJBL*

Component	Before PJBL implementation	With PJBL implementation
Course management	4,36 (fairly satisfied)	4,82 (satisfied)
Teaching and learning method	4,28 (fairly satisfied)	4,96 (satisfied)
Overall satisfaction	4,29 (fairly satisfied)	4,90 (satisfied)
Total average	4,31 (fairly satisfied)	4,89 (satisfied)

The data in the table shows that with the implementation of PJBL in the Introduction to Programming course, students' satisfaction with the learning experience increased. This implies that the new course fulfills the needs of learning from the results of reflection phase indicators (satisfaction of learning process) from the PJBL implementation in Introduction to Programming.

In this first cycle, the researchers found that the implementation of PJBL in the Introduction to Programming course has been proven to improve programming competency and students are satisfied with the learning process. Based on the research, we concluded that PJBL can be implemented in the Introduction to Programming course with improvement in the retention of knowledge.

Findings and Discussion

This action research enriches the implementation of PJBL in the computer science field of study. Most research on PJBL looks at and tries to explain the effectiveness of PJBL in increasing the learning result in mastering specific skills. In those fields, much research shows that PJBL is effective in improving learning results and providing meaningful learning through the experiential learning approach of PJBL (Efstratia, 2014; Yazici, 2020).

This research examines the design, implementation, and evaluation of PJBL in the field of computer science, especially in increasing programming skills. The expectation of implementing PJBL in the undergraduate program is that PJBL can also support students in increasing some intrinsic skills needed in the computer science field, such as problem-solving skills, self-regulated learning, collaborative learning, as well as lifelong learning. The authors tried to answer the following research questions: a) how to implement PJBL in a programming course to improve programming competency, and b) how to ensure students' satisfaction in the learning process.

Some research, such as that by Yazici (2020), explains how PJBL can increase the learning result. Yazici (2020) explained the application of PJBL to business undergraduate students and found that there is an increase in the student's retention of understanding and improved transfer of skills through business problems (Yazici, 2020). Our research also found the same result—PJBL can be effective in increasing learning. We found not only that PJBL increases learning, but also that PJBL helps students transfer their programming skills into real-life projects. However, this research also notes that specific intervention needs to be conducted to ensure the retention of programming knowledge since the students' retention level of programming knowledge still needs improvement. To remind that students will actively construct their own knowledge by executing their project, we found that at some point the constructed knowledge can be misleading. Therefore, a specific intervention should be done by the facilitator to ensure that the constructed knowledge is on the right track.

This research also enriches the research in PJBL that shows how PJBL also supports intrinsic skills such as self-regulated learning, critical and analytical thinking, as well as a passion for lifelong learning (English & Kitsantas, 2013; Isomottonen et al., 2019; Yazici, 2020). As this research found an interesting result that there is also an increase in students' engagement in that first-semester student cohort. The increase in engagement was due to the fact that the PJBL format for the course alters how the class was taught; the class was taught in a more intensive manner for 3 days a week. This allows for more discussions and opportunities for having more questions answered. Engagement also increased since all assignments had to be completed during the day; students are pressured to experiment, learn, and solve problems through search and discussion with classmates. That learning process makes the course memorable for students and increases engagement between students as well as with the learning process itself.

Additionally, the level of confidence of students in programming and software development has also increased. This is because PJBL allows for more discussion, collaboration, and exploration. Students gained the habit and became comfortable with trying new concepts and tools that are not necessarily taught or even covered within the course outline or syllabus. As an example, advanced programming concepts such as implementing graphics, multimedia, and sound were common in the work done by students, although it was not a requirement. In several cases, a few adventurous students even implemented advanced concepts such as network programming since they are challenged to develop programs that can communicate with gadgets that they own at home. Students were able to search and learn to implement concepts from their exploration on the internet. This indicates that their metacognition capability in learning also improved; so that they can maintain and control the speed and rhythm of learning from various learning sources.

The findings in this research enrich research related to PJBL, specifically, PJBL in computing. The authors believe that PJBL is a learning method that can be implemented in education at any level or field of study. Since PJBL has such potential for different areas of learning, we suggest that research on PJBL should continue and specifically deeply explore three areas: a) improving learning outcomes, specifically how knowledge retention of students can be improved, b) learning interactions, better understanding of self-regulated learning that occurs in PJBL, and c) development of alternative learning sources, specifically sources of digital learning that are effective for use in PJBL.

References

Bender, W. N. (2012). *Project-based learning: Differentiating instruction for the 21st century*. Corwin Press.

Brendel, W., & Cornett-Murtada, V. (2019). Professors practicing mindfulness: An action research study on transformed teaching, research, and service. *Journal of Transformative Education*, *17*(1), 4–23. https://doi.org/10.1177/1541344618762535

Efstratia, D. (2014). Experiential education through project based learning. *Procedia—Social and Behavioral Sciences*, *152*, 1256–1260. https://doi.org/10.1016/j.sbspro.2014.09.362

English, M. C., & Kitsantas, A. (2013). Supporting student self-regulated learning in problem- and project-based learning. *Interdisciplinary Journal of Problem-Based Learning*, *7*(2). https://doi.org/10.7771/1541-5015.1339

Isomottonen, V., & Nylen, A. (2019). Multiple authentic project-based experiences and persistent learning? *49th IEEE Frontiers in Education Conference, FIE 2019*, *2019–October*, pp. 1–5. https://doi.org/10.1109/FIE43999.2019.9028535

Kokotsaki, D., Menzies, V., & Wiggins, A. (2016). Project-based learning: A review of the literature. *Improving Schools, 19*(3). https://doi.org/10.1177/1365480216659733

Lee, C. W. Y., Chu, S. K. W., Cheng, J. O. Y., & Reynolds, R. (2016). Plagiarism-free inquiry project-based learning with UPCC pedagogy. *Proceedings of the Association for Information Science and Technology*, *53*(1), 1–11. https://doi.org/10.1002/pra2.2016.14505301033

Montero, C. S. (2018). Craft- and project-based making for STEAM learning. *18th Koli Calling Conference on Computing Education Research, Koli Calling 2018*. https://doi.org/10.1145/3279720.3289237

PBLWorks. (2022, February 5). *Gold Standard PBL: Essential Project Design Elements*. https://www.pblworks.org/what-is-pbl/gold-standard-project-design

Thomas, J. (2000). A review of research on project-based learning. In *Learning*. https://www.researchgate.net/publication/238162544_A_Review_of_Research_on_Project-Based_Learning

Yazici, H. J. (2020). Project-based learning for teaching business analytics in the undergraduate curriculum. *Decision Sciences Journal of Innovative Education*, *18*(4), 589–611. https://doi.org/10.1111/dsji.12219