GAMIFICATION OF CAT-IRT FOR SELECT DATA STRUCTURES CONCEPTS TO ENHANCE STUDENT'S PROGRAMMING UNDERSTANDING

BY

AKINSANYA, ADEYINKA OLASENI
(21CG029820)

A PROJECT PROPOSAL SUBMITTED TO THE DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES, COLLEGE OF SCIENCE AND TECHNOLOGY, COVENANT UNIVERSITY, OTA, OGUN STATE.

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF SCIENCE (HONOURS) DEGREE IN COMPUTER SCIENCE.

JUNE, 2025

CERTIFICATION

I hereby certify that this project was carried out by Akinsanya, Adeyinka Olaseni in the department of Computer and Information Sciences, college of Science and Technology, Covenant University, Ota, Ogun State, Nigeria under my supervision.

Dr. Olamma Iheanetu
Supervisor

____
Signature and Date

Prof. Aderonke Oni
Head of Department

____
Signature and Date

DEDICATION

I dedicate this work to God, my constant source of strength and support throughout my four-year journey in this institution. I also dedicate it to my beloved parents, Dr. Adebola Akinsanya and Mrs. Omolade Akinsanya, whose unwavering support and encouragement have meant so much to me.

ACKNOWLEDGEMENTS

I am deeply thankful to God for His unwavering guidance and support throughout my educational journey. I am sincerely grateful to Dr. Adebola Akinsanya and Mrs. Omolade Akinsanya for their endless support and heartfelt prayers. I also extend my heartfelt appreciation to everyone who contributed to the success of this study, especially my supervisor, Dr. Olamma Iheanetu, for her invaluable guidance.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API	Application Programming Interface
BOBCAT	Bilevel Optimization-Based Computerized Adaptive Testing
CAT	Computerized Adaptive Testing
CSV	Comma Separated Values
IRT	Item Response Theory
nDGR	Normalized Discounted Cumulative Gain
MRR	Mean Reciprocal Rank
RMSE	Root Mean Square Error

ABSTRACT

The work centers on building an adaptive gamified test platform that enhances learners' comprehension of specific data structures topics. Through the use of an existing CAT-IRT model called BOBCAT, the system chooses questions based on each individual's performance level while providing an individualized test experience. The BOBCAT model was trained using a manually selected set of multiple-choice questions with IRT parameters like difficulty level, discrimination level, and probability of guessing.

An interaction dataset having 500 virtual students, each answering 30 questions, was created using a Python script to replicate realistic training data. The model was able to understand how students with varying skill levels respond to different kinds of questions thanks to these interactions. To have the data ready for training, categorical parameters were also transformed into numerical scales.

A full-stack methodology was used in the platform's construction. Next.js was used to construct the frontend, and Node.js and TypeScript were used to implement the backend. Users, test sessions, questions, answers, and gamified components like leaderboards and badges were all stored in a MongoDB database. Communication between the adaptive engine and the frontend was made easier by RESTful APIs.

Using common measures like nDCG, MRR, and RMSE, the model's performance was assessed. All things considered, the system combines gamification and adaptive assessment to provide an innovative and interesting learning experience. It illustrates how individualized learning resources can improve student's conceptual comprehension of programming, especially when it comes to data structures.

CHAPTER ONE

INTRODUCTION

1.1. Background Information

Programming has become a crucial ability in the digital age for a variety of fields, including data analytics, artificial intelligence, and software development. However, many students still find programming to have a high learning curve, especially when dealing with abstract ideas like data structures. The varied learning requirements and motivational levels of contemporary students may not be sufficiently satisfied by traditional teaching approaches, which frequently rely on lectures and textbook exercises.

A promising tactic to raise student interest and comprehension is gamification, which is the incorporation of game aspects into non-gaming environments. Numerous studies highlight its effectiveness in enhancing engagement and motivation across educational domains. For example, Lavoué *et al.* (2021) found that gamification fosters achievement-oriented and perfection-oriented behaviours, boosting motivation and persistence in learning tasks. Gaming may change how programming topics are taught and understood by introducing interactive challenges, rewards, and immersive experiences. Data structures like stacks, queues, trees, and graphs in particular need a solid conceptual underpinning, which games' visual and hands-on learning techniques can help to reinforce.

At the same time, using intelligent testing technologies has become essential for more accurate evaluation of student achievement. Item Response Theory (IRT)-powered Computerized Adaptive Testing (CAT) provides a dynamic method of assessing students' comprehension by customizing questions according to their proficiency level. By lowering or increasing the quality of questions required to gauge performance (Ma et al., 2023), CAT-IRT not only improves assessment accuracy but also lessens test fatigue.

Since data structures are widely acknowledged as the cornerstone of programming and computers, I selected it as the subject of this adaptive testing platform. Effective data storage, manipulation, and retrieval are made possible by data structures and are fundamental processes that support almost every branch of computer science.

The core components of computing are data structures, which are necessary for developing and putting into practice algorithms that can process vast volumes of data rapidly and precisely (Dwaraka Srihith et al., 2023).

Although gamification has demonstrated promise in enhancing learning outcomes and motivation in educational contexts, its potential in programming education, particularly when paired with adaptive assessment techniques, has not yet been thoroughly investigated. Similarly, computer science education continues to underutilize the incorporation of Computerized Adaptive Testing based on Item Response Theory (CAT-IRT) into game-based learning settings.

In order to accurately measure and support individual learning progress, it is necessary to develop and assess a learning framework that not only improves learners' comprehension of data structures through interactive and engaging methods, but also integrates clever, adaptive assessment techniques.

This work demonstrates how a game-based approach will improve students' understanding of data structures through CAT-IRT, hence examining the junction of gamification and adaptive assessment. The goal of the research is to close the gap between effective assessment and student engagement in programming education by combining adaptive testing techniques with gamification techniques.

1.2. Statement of Problem

Even though programming skills are becoming more and more important in today's technologically advanced world, many students still have trouble grasping fundamental computer science concepts, especially data structures, which are abstract, logical, and frequently call for deep analytical thinking. Consequently, learners often show low retention rates, poor engagement, and limited problem-solving skills when it comes to applying their knowledge of data structures in real-world situations.

Additionally, the techniques employed to evaluate students' programming comprehension are frequently strict and uniform, ignoring the variations in individual learning styles (Wiredu et al., 2024). Traditional tests might overwhelm struggling students or demotivate high-achieving students since they don't adjust to the skill level of the learners. This one-size-fits-all strategy could lead to assessments of student’s true competencies that are not accurate.

While gamification has shown promise in enhancing engagement and motivation in educational contexts, its application to computing education remains limited and often lacks depth. Many gamified platforms focus on introductory programming or computational thinking but fail to address the more advanced and abstract topics central to computing curricula. Additionally, these tools often prioritize entertainment over rigorous, curriculum-aligned learning. This gap presents an opportunity to leverage gamification principles to create an innovative, engaging, and effective learning platform tailored specifically to the needs of computing students (Triantafyllou et al., 2024).

1.3. Aim and Objectives of the Study

The aim of this research is to design and evaluate a gamified web application to enhance the understanding of data structures using adaptive testing. The application will integrate gamification principles to foster engagement, motivation, and improved learning outcomes among computing students.

The research is guided by the following key objectives:

- **i.** To curate a dataset of data structure questions.
- **ii.** To design a database for the gamified adaptive testing platform.
- **iii.** To fine tune a pre-existing CAT-IRT (BOBCAT) model for adaptive testing.
- **iv.** To develop the interface of the platform.
- **v.** To implement and integrate the adaptive testing in the platform.
- **vi.** To evaluate the proposed model using evaluation metrics.

1.4. Research Methodology

This study makes use of a mixed-methods approach, using both exploratory design and applied design approach. The need for a better CAT-IRT model is determined and then the validated model is integrated into a functional web-based adaptive testing platform with gamification components. Table 1.1 shows the mapping of each objective to a method:

Table 1.1: Mapping of Objectives to Methods

S/N	Objective	Methodology	Tools/ Techniques
1.	To curate a dataset of data structure questions.	Using python script to scrape standardized data structures questions.	Python.
2.	To design a database for the gamified adaptive testing platform.	Using MongoDB and Mongoose to define database tables (collections) and schemas.	MongoDB for storing data, Mongoose for database design
3.	To fine tune a pre-existing CAT-IRT (BOBCAT) model for adaptive testing.	Using a carefully curated dataset of annotated test questions arranged by Item Response Theory Parameters to fine-tune a CAT model.	Python, Hugging Face Transformers, PyTorch/TensorFlow, annotated datasets.
4.	To develop the interface of the platform.	Using web development frameworks to develop the frontend interface	Next.js, Typescript.
5.	To implement and integrate the adaptive testing in the platform.	Developing APIs for integration and integrating the model	Render, Flask, Express.js, Typescript.

		into the platform's backend	
6.	To evaluate the proposed model using evaluation metrics.	Evaluation metrics like Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (nDCG), Root Mean Square Error (RMSE) to measure model's performance.	PyTorch, pandas, numpy, sklearn.

1.5. Significance of the Study

Both academically and technologically, this work is extremely relevant, especially when it comes to teaching programming. For students learning data structure concepts, the research helps create more efficient, interesting, and customized learning environments by combining gamification with an adaptive testing technique, Computerized Adaptive Testing based on Item Response Theory (CAT-IRT).

This study offers students a fresh and captivating method of understanding intricate programming principles through interactive, game-based instruction. Test anxiety is lessened and learners' confidence in their learning process is increased because the assessment's adaptive nature guarantees that students are assessed at their appropriate ability levels.

The study's findings can help curriculum designers and institutions create creative and successful programming courses that are both engaging and in line with contemporary educational standards. The incorporation of CAT-IRT into gamified platforms can result in more accurate assessments of student capabilities, supporting data-driven decision-making in curriculum development. For educators, the research offers insights into how gamification and adaptive testing can be used to enhance teaching strategies and learning outcomes. It also provides a framework.

This study adds to the expanding corpus of research on the nexus of technology, pedagogy, and evaluation in the field of computer science education. It illustrates how adaptive testing and game dynamics can work together to solve persistent problems in programming education, which may have an impact on further study and use in other STEM domains.

By using creative and student-centred methods, the study hopes to close the gap between engagement and successful assessment, significantly advancing the reform of programming education.

CHAPTER TWO

LITERATURE REVIEW

2.1. Preamble

In the context of data structure concepts, this chapter provides a thorough analysis of the literature that is pertinent to improving students' comprehension of programming through the integration of gamification and adaptive assessment systems. The chapter focuses on the following: a review of core concepts relating to gamification and adaptive testing in programming education, related testing methods and a review of existing publications that support the use of adaptive testing and gamification. This aim of this literature review is to find out the gaps in existing research and to provide a theoretical basis for this project.

2.2. Review of Relevant Concepts

Understanding the fundamental ideas that support the use of gamification and adaptive testing in programming education is crucial to building a solid foundation for this research. This section goes over important concepts and words such data structures, gamification, computerized adaptive testing (CAT), item response theory (IRT), and programming education.

2.2.1. Programming Education

Programming has aspects of good design that make it an excellent learning experience, such as the iterative process towards a solution, the ultimate goal that has many paths and choices on the way, and the mistakes that provide feedback and experience. Good student-teacher relationships, clear explanations of important concepts, students working at their own pace, meaningful exercises, games, challenges, and competition to spark interest, boosting learner confidence, and making sure students feel comfortable asking for help are just a few of the important elements that research has found can support programming learner motivation(Boguslawski et al., 2024).

Teaching students to develop, evaluate, and optimize code using programming languages and tools is known as programming education. It places an emphasis on reasoning, problem-solving, and computational thinking. However, the abstract nature of programming makes it difficult for many students, particularly when they are exposed to complicated concepts like data structures. As a result, there is now a need for more individualized and interactive teaching methods.

2.2.2. Data Structures

Because of their abstract nature, data structures are often seen as challenging by novices, requiring alternative teaching strategies that enhance visualization and comprehension. Data structures play a vital role in enhancing software or program performance by optimizing the way data is stored and retrieved. The main objective of a data structure is to facilitate efficient and effective data operations. Correctness, which ensures that its implementation accurately complies with the defined interface; time complexity, which refers to minimizing the execution time of operations; and space complexity, which focuses on minimizing the memory usage during data operations, are important features of a data structure (Hasan, 2024).

2.2.3. Gamification

Gamification is the process of applying game features like leaderboards, challenges, points, levels, and feedback loops to non-gaming contexts in order to boost user motivation and engagement. Gamification in education has demonstrated benefits in raising student engagement, retention, and interest. It can simulate real-world problem-solving, offer instant feedback, and foster an interactive learning environment when used in programming.

The deliberate incorporation of game design features, mechanisms, and concepts into non-gaming contexts is known as "gamification" or "gameful design." It is frequently enabled through digital platforms with the purposes of problem-solving, engagement, and goal-motivation. By encouraging a playful and participatory experience, the method improves users' perceptions of their own independence, skill, and relatedness. Gamification is a flexible technique that enhances user experience and adds value in a variety of contexts, with origins in industries including education, business, marketing, and services. Gamification's pervasive influence in many fields has changed conventional engagement strategies, most notably in education (Ružic & Dumancic, 2015).

2.2.4. Computerized Adaptive Testing (CAT)

CAT is a dynamic assessment technique that instantly modifies the test items' level of difficulty in response to a test-taker's answers. In contrast to conventional testing, CAT chooses questions based on the learner's proficiency level, improving assessment efficiency and accuracy. CAT maintains excellent measurement precision while lowering the number of questions required to assess proficiency.

Computerized adaptive testing (CAT) targets to accurately assess the student's Proficiency in the required subject/area. The key issue is how to design a question selector that adaptively selects the best-suited questions for each student based on previous performance step by step (Ma et al., 2023).

2.2.5. Item Response Theory (IRT)

IRT is a statistical paradigm used in test design and analysis, focused on the relationship between a learner's latent capacity and the chance of answering a question correctly. It forms the mathematical backbone of CAT, enabling the system to select topics that are properly hard. IRT takes item difficulty, discrimination, and guessing criteria into account to provide individualized testing experiences.

Educational assessment experts are increasingly using item response theory to analyse cognitive measurement, but little is known about the theory to analyse noncognitive measurement. Item response theory is a measurement framework used in the design and analysis of educational and psychological assessments (achievement tests, rating scales, inventories, or other instruments) that measure mental traits (Ogunsakin & Shogbesan, 2018).

2.3. Review of Related Methods

In this section, we evaluate the methodologies and strategies that have been implemented in the literature to enhance programming education, with a particular emphasis on gamification and adaptive testing approaches. These strategies are important in understanding how educational technologies can be utilized to improve learning results, particularly in complicated areas like data structures.

2.3.1. Traditional Methods in Programming Education

Traditional programming education approaches usually involve lectures, textbooks, and static assignments. Although these approaches work well for foundational learning, they frequently do not involve students in active problem-solving or offer timely feedback. Additionally, traditional assessments are typically standardized and do not take into account the needs or styles of individual students, which can cause dissatisfaction, disengagement, and inaccurate assessments of student ability.

2.3.2. Gamified Learning Approaches

Gamification is defined as the integration of game mechanics, such as points, badges, leader boards, and rewards, into non-gaming environments to enhance user engagement and achieve desired outcomes. In the educational context, gamification has garnered significant attention for its ability to foster motivation, engagement, and improved learning outcomes across diverse disciplines. Its impact has been widely documented in numerous studies, underscoring its versatility and efficacy.

Lavoué *et al.* (2021) demonstrated that gamification promotes achievement-oriented and perfection-oriented behaviours among learners, significantly enhancing their motivation and persistence. This aligns with Jaramillo-Mediavilla *et al.* (2024), whose systematic review showed that thoughtfully designed gamified learning environments lead to substantial improvements in student motivation, self-directed learning capabilities, and academic performance. Together, these findings highlight the transformative potential of gamification when applied effectively.

Karsen *et al.* (2022) emphasized the role of gamification in reducing attrition rates in Massive Open Online Courses (MOOCs) by leveraging mechanisms such as badges and reward systems to increase student participation and motivation. Their work underscores the importance of tailoring gamification strategies to align with the diverse needs of learners. Similarly, Rizzardini *et al.* (2016) found that gamification elements such as progress tracking and rewards significantly enhance student engagement, provided they are aligned with learning objectives. These findings form the basis for the inclusion of adaptive mechanics in this project's design to accommodate varied learner profiles.

Moreover, gamification is not only effective in fostering a sense of accomplishment but also in creating an enjoyable learning experience that encourages sustained engagement. However, to maximize its benefits, it is crucial to balance motivational elements with the cognitive demands of the learning material. Overuse of competitive features, for instance, can lead to unintended stress and disengagement, necessitating careful and thoughtful design choices.

Gamification incorporates a variety of elements to enhance learning experiences. Each gamification element contributes uniquely to the overall impact, ensuring a more engaging and effective educational process. Below is an extensive discussion of the key elements:

i. Points and Scoring

Points and scoring systems are foundational elements of gamification that provide learners with immediate feedback and a tangible sense of progress. By assigning points for completing tasks, answering questions correctly, or achieving specific milestones, these systems incentivize participation and reward effort.

The psychological impact of earning points is significant. Research shows that points trigger the brain's reward system, releasing dopamine and creating positive reinforcement (Lavoué et al., 2021). This immediate gratification motivates learners to continue engaging with the material, fostering a sense of accomplishment and driving consistent participation.

Moreover, points serve as a metric for tracking progress, allowing learners to visualize their improvement over time. This visibility can be particularly motivating for students in challenging subjects, as it highlights incremental achievements that might otherwise go unnoticed. Teachers and educators can also use points as a diagnostic tool to identify areas where students may need additional support.

However, the design of point systems must be carefully balanced to avoid unintended consequences. For instance, overemphasis on point accumulation can lead to extrinsic motivation, where learners focus solely on earning points rather than mastering the subject matter. To address this, educators can combine points with other gamification elements, such as meaningful feedback and rewards, to maintain intrinsic motivation.

Finally, points and scoring systems can be integrated into adaptive learning environments, where tasks are adjusted based on performance. This ensures that learners are continuously challenged at an appropriate level, maximizing engagement and knowledge retention.

ii. Badges and Achievements

Badges and achievements are visual representations of accomplishments that recognize learners’ efforts and milestones. These elements foster a sense of pride and progression, motivating students to pursue further achievements and complete tasks.

Badges often serve as markers of specific competencies or skills, such as mastering a programming concept or completing a challenging algorithm exercise. This recognition not only boosts learners' confidence but also provides them with tangible evidence of their progress. Employers and institutions can also use badges as indicators of proficiency in specific areas (Rizzardini et al., 2016).

The psychological benefits of badges extend beyond individual motivation. Studies show that public displays of achievements, such as digital badges visible on leaderboards or profiles, encourage friendly competition and collaboration among peers (Oliveira et al., 2022). This social aspect of gamification can create a supportive learning community, where learners motivate each other to succeed.

However, the effectiveness of badges depends on their relevance and alignment with educational objectives. Badges that are too easy to earn may lose their value, while overly complex or ambiguous criteria can discourage learners. Therefore, it is essential to design badges that strike a balance between accessibility and challenge.

Incorporating hierarchical achievements, where learners unlock advanced badges by building on foundational skills, can further enhance the learning experience. This approach encourages sustained engagement and progression, guiding learners through increasingly complex material.

iii. Leaderboards

Leaderboards rank learners based on their performance, fostering a sense of competition and collaboration. By displaying rankings publicly or within a group, leader boards create an environment where students are motivated to improve their standing.

The competitive nature of leaderboards can be a powerful motivator for learners, particularly in subjects that require sustained effort and practice. For example, students might strive to solve more programming challenges or complete assignments faster to climb the leaderboard. This dynamic encourages active participation and continuous improvement (Li et al., 2024).

In addition to individual motivation, leaderboards can promote collaboration. Group leaderboards, where teams compete against each other, foster teamwork and communication skills. This collaborative aspect is particularly valuable in computing education, where problem-solving often involves working with others to develop solutions.

Despite their benefits, leaderboards must be implemented thoughtfully to avoid negative effects. Overemphasis on rankings can create stress or discourage learners who struggle to compete with high performers. To mitigate this, educators can use tiered leaderboards, which group learners into categories based on skill level. This approach ensures that all students can experience the benefits of competition without feeling overwhelmed.

Furthermore, integrating leaderboards with other gamification elements, such as points and badges, can enhance their effectiveness. For instance, displaying badges alongside rankings can highlight individual strengths and achievements, fostering a more inclusive and supportive learning environment.

iv. Rewards and Incentives

Rewards and incentives are integral to gamification, providing learners with tangible or intangible benefits for their achievements. These elements create a sense of anticipation and satisfaction, motivating students to engage with the material and complete tasks.

Rewards can take various forms, such as virtual items, certificates, or even privileges within the learning platform. For example, learners might earn tokens that can be exchanged for hints or additional resources. These incentives add an element of fun and excitement to the learning process, making it more enjoyable and memorable (Troussas et al., 2019).

The timing and delivery of rewards are crucial to their effectiveness. Immediate rewards, such as feedback or small tokens, reinforce positive behaviour and encourage learners to continue their efforts. On the other hand, delayed rewards, such as completing a course or achieving a major milestone, provide a sense of accomplishment and closure.

While rewards are effective in driving engagement, they should be used judiciously to avoid over-reliance on extrinsic motivation. Balancing rewards with intrinsic motivators, such as personal growth and mastery of skills, ensures that learners remain committed to their educational goals beyond the gamified elements.

Lastly, incorporating personalized rewards, tailored to individual preferences and achievements, can enhance their impact. For instance, allowing learners to choose their rewards or setting personalized goals adds a layer of autonomy and relevance to the gamified experience.

2.3.3. Adaptive Testing Methods

Computerized Adaptive Testing (CAT), in particular, has been used more and more in educational settings to provide more accurate and personalized assessments. CAT uses Item Response Theory (IRT) to measure student abilities more precisely by taking into account factors like question difficulty, discrimination, and guessing. CAT dynamically selects questions based on the learner's ability, minimizing test fatigue and optimizing the learning experience.

Research has demonstrated that by accommodating individual differences, CAT improves test validity while simultaneously lowering the number of items required to evaluate a learner's competency. Adaptive testing can be very helpful in programming education since it allows examinations to be customized for students with varying ability levels, giving a more accurate picture of their understanding of data structures and other programming principles.

Many educational testing centres worldwide are very interested in CAT's ability to create tests that are specific to the ability levels of examinees. A number of testing organizations have chosen to employ CAT in place of more conventional paper-and-pencil exams (Latu & Chapman, 2002).

2.3.4. Combining Gamification and Adaptive Testing

Learning methodologies, adaptive aspects, user kinds, and motivational theories are all linked to the fundamental ideas of adaptive gamification. According to the pertinent literature, there aren't many known adaptive gamification strategies, and even fewer that deal with particular material. Because motivating theories take into account the connection with students' needs and features and are closely related to learning methodologies in science education, developing an adaptable gamification environment has proven to be a substantial problem (Zourmpakis et al., 2024).

Although adaptive testing and gamification have been studied independently, recent work has concentrated on combining these techniques to produce more individualized and captivating learning environments. Combining game elements with CAT gives the ability to deliver dynamic assessments that correlate with learners' progress in real-time while keeping the incentive aspects of gamified environments.

Hybrid systems that combine gamification and adaptive assessments have been studied in a few studies. These systems usually provide students with a game-like interface where they can unlock levels or earn rewards based on how well they perform in adaptive tests. For instance, a learner may finish a series of data structure challenges, and the system adjusts the questions to gradually match their ability level as they succeed. This creates a personalized learning experience that keeps students motivated while accurately measuring their comprehension of programming concepts.

Blended learning mixes online and traditional classroom training, aiming to enhance educational outcomes. Notwithstanding its promise, student involvement in Online components continue to be a significant obstacle. Although its effectiveness is debatable and research has produced conflicting findings, gamification has been a popular strategy to increase engagement (Zhang & Huang, 2024).

Adaptive learning tailors educational experiences to individual learners by adjusting content and difficulty levels based on user performance. This approach ensures that learners remain appropriately challenged, fostering continuous engagement and growth. Hooshyar *et al.* (2020) demonstrated that adaptive educational games improve engagement by dynamically adjusting challenges to match learner capabilities, a principle central to the proposed project's design.

The importance of adaptability is further emphasized by Troussas *et al.* (2019), who investigated the use of fuzzy-modelled personalization in mobile game-based learning. Their findings indicate that tailored feedback and personalized recommendations significantly enhance learning experiences, supporting the integration of similar adaptive mechanisms into the proposed platform. Adaptive gamification not only sustains engagement but also promotes a growth mindset by encouraging learners to tackle progressively challenging tasks without feeling overwhelmed.

2.3.5. Challenges in Implementing Gamification and Adaptive Testing

Science education can be enhanced by combining gamification apps with contemporary teaching methods including inquiry-based learning, flipped learning, and problem-solving learning. Studies have indicated that learning results are enhanced when these tactics are combined with gamification. The research gap in this area can be explained by the fact that implementing various learning strategies necessitates significant adjustments to the gamification environment, which may limit their viability (Zourmpakis et al., 2024).

Gamification and adaptive testing have a lot of promise, but putting them into practice can be difficult. The requirement for complex algorithms and technology to adaptively choose questions while striking a balance between difficulty and success is one of the fundamental challenges. Furthermore, it's important to carefully evaluate how to keep students engaged without overloading them with assessments when integrating gamified features with adaptive testing.

The creation of engaging and instructive content presents another difficulty. While games can provide immersive settings, they must be properly developed to ensure that the learning experience is not overpowered by the entertainment side. Additionally, even if adaptive testing is effective, it needs to be improved constantly to guarantee that the questions chosen are accurate and trustworthy indicators of student aptitude.

2.4. Review of Existing Papers

In this section, I review earlier studies on the use of Item Response Theory (IRT) to gamify Computerized Adaptive Testing (CAT), specifically in the context of teaching programming. Studies that investigate the application of concept-specific techniques, adaptive testing frameworks, and gamified learning environments for fundamental programming topics particularly data structures are highlighted. Finding research gaps and demonstrating the value of using CAT-IRT to choose data structure ideas as a way to enhance students' programming knowledge are the objectives.

The following are the reviewed papers on existing works in this domain:

2.4.1. Computer-Adaptive Testing in Science Education, by Elena C. Papanastasiou.

The use of computer-adaptive testing (CAT) to improve the efficacy of computer-based learning (CBL), especially in science education, was examined by (Papanastasiou, 2003). The study provides a thorough description of how CAT, which is based on Item Response Theory (IRT), dynamically modifies question difficulty in real-time to correspond with each examinee's predicted skill level. This method reduces the number of items given while enabling a more accurate assessment. Increased item efficiency, quicker testing periods, instant feedback for teachers and students, and more adaptable item formats made available by multimedia integration are just a few advantages of CAT that are discussed in this study. She also highlights how the system's entertaining exam interfaces can boost motivation and accommodate kids with unique needs.

Although (Papanastasiou, 2003), in her work offers a strong basis for comprehending the advantages and difficulties of CAT's implementation in science education, it mainly concentrates on classroom assessment and theoretical adaptation rather than the incorporation of cutting-edge computational tools or gamification techniques. No gamified components, machine learning-based model fine-tuning, or customized data structure material targeted at enhancing programming comprehension are included in her study.

By applying CAT-IRT specifically to data structure principles in computer science education, the current study expands on the idea. It does this by combining a gamified frontend with a refined adaptive model. This difference highlights the innovation and technical work behind the platform, which uses interactive, data-driven, and adaptive learning experiences to actively improve programming comprehension in addition to assessing it.

2.4.2. Computer-Adaptive Testing for Students with Special Educational Needs, by Ebenbeck and Nikola

This thorough study looked into the use of computerized adaptive testing (CAT) to improve assessment procedures for slow learners and children with special educational needs (SEN). In contrast to conventional assessment methods, the study shows how CAT technology can offer more efficient and customized measures by employing adaptive algorithms that continuously re-estimate examinee ability and choose the best next items. The work demonstrates that CAT provides notable benefits for diverse ability groups through three simulation experiments based on a sample of 400 kids (22.5% with SEN, including those with learning disabilities, intellectual disabilities, and speech impairments). The results show that adaptive tests are 30-80% shorter than non-adaptive versions, and children with SEN complete more accurate assessments with about 4 fewer items than their non-SEN peers. The study offers important insights into the best ways to deploy CAT in inclusive educational contexts by examining different halting rules, starting methods, and item pool designs. However, this study mainly focuses on reading assessment within conventional educational evaluation frameworks rather than investigating integration with modern educational technologies or cutting-edge pedagogical approaches, even though it provides strong empirical evidence for CAT's efficacy in SEN assessment and offers comprehensive technical recommendations for adaptive screening development. The study omits components that could improve the adaptive testing experience even more, like web-based interactive platforms, machine learning-based model fine-tuning, and gamified learning interfaces. In order to create a more comprehensive adaptive learning ecosystem that not only assesses but also actively supports diverse learners through personalized, game-enhanced programming instruction and real-time adaptive question delivery, the current research builds on this foundational work by applying CAT principles to computer science data structures education. It does this by integrating gamification elements with a refined BOBCAT model and an interactive web-based platform. This contrast highlights how holistic, technologically enhanced learning environments that take advantage of contemporary computing capabilities to support students with a range of educational needs have replaced traditional adaptive assessment (Ebenbeck, 2023).

2.4.3. Applications of Computerized Adaptive Testing in Educational Settings, by Weiss and Kingsbury

Through three different applications, the use of computerized adaptive testing (CAT) to meet certain educational decision-making difficulties was thoroughly investigated. In contrast to conventional fixed-length paper-and-pencil tests, the study shows how CAT technology, which is based on Item Response Theory (IRT), can dynamically select test questions from item pools to predict student achievement levels with greater precision. Through empirical research on adaptive self-referenced testing, adaptive grading, and adaptive mastery testing, the work demonstrates that CAT processes use up to 50% less test questions than traditional methods while achieving greater accuracy in mastery classifications. According to the results, adaptive tests continue to have validities and reliabilities that are on track with or higher than those of similar conventional tests, showing particular efficiency in military testing settings. The study's analysis of IRT-based adaptive testing techniques offers important new information on how to administer computer-interactive tests and estimate trait levels after individual item answers. Nevertheless, this seminal study mainly concentrates on conventional educational assessment paradigms rather than investigating integration with contemporary teaching or learning methodologies, even though it offers a strong theoretical framework for CAT implementation in educational settings and convincing evidence of adaptive testing's effectiveness in ability measurement contexts. The study excludes components that could improve the adaptive testing experience for specialized topic areas, such as web-based interactive learning platforms, domain-specific educational content delivery, and gamified assessment interfaces. In order to create a more comprehensive educational ecosystem that not only assesses programming competency but also actively supports conceptual understanding through interactive, game-enhanced learning experiences and personalized adaptive question pathways tailored to programming education, the current research builds on this innovative study by applying CAT-IRT principles to computer science education, specifically data structures instruction. This contrast highlights how specialized, technologically enhanced learning environments that use modern computational capabilities and educational gaming ideas to support students in technical fields have replaced traditional adaptive assessment (Weiss & Kingsbury, 1984).

2.5. Summary and Conclusion

This chapter offers a thorough assessment of research on computerized adaptive testing and its uses in education. Through customized, dynamic testing methods, the investigation looked at how CAT technology, which is based on Item Response Theory, improves educational evaluation.

Significant fundamental contributions in a number of educational circumstances are revealed by the review. The potential of CAT for real-time difficulty adaption and multimedia integration in science education was illustrated by Papanastasiou's (2003) work. According to Ebenbeck and Nikola's research, adaptive assessments can reduce test length by 30 to 80% while still retaining high accuracy, which is especially advantageous for students with special education requirements. According to Weiss and Kingsbury's groundbreaking research, CAT techniques can produce better mastery classifications with up to 50% less inquiries than traditional methods.

However, these innovative studies did not investigate contemporary educational tools, instead concentrating on conventional educational frameworks. Although these studies had strong theoretical underpinnings, they did not incorporate web-based interactive platforms, gamified learning interfaces, or machine learning-based model fine-tuning, which could improve the adaptive testing experience for specific topic areas.

This review of the literature identifies a crucial gap, although CAT technology has demonstrated remarkable efficacy in a number of fields, there is plenty of room to incorporate adaptive testing concepts with contemporary educational technologies, especially for specialized subjects that call for conceptual understanding.

By expanding on these theoretical foundations and filling in the identified technological gaps, this study applies the concepts of adaptive testing to computer science education, with a focus on teaching data structures. Through customized, game-enhanced learning experiences designed for programming education, the suggested method actively promotes conceptual understanding in addition to evaluating programming competency. For technical fields, this signifies a shift from conventional adaptive assessment to specialized, technologically enhanced learning settings.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1. Preamble

This chapter covers the research methodology used for this study. It outlines the research design used, data collection, database design, model fine tuning, the tools and methodologies utilized, and the criteria for measurement of the effectiveness and performance of the model. The process of the methodological design has been organized around iterative system design and experimental evaluation.

3.2. Research Design

To design a gamified adaptive testing platform to improve the understanding of data structure concepts among students, this study adopts a two-phase design strategy integrating an exploratory with an application technique. To enhance performance and interest, the platform utilizes gamification elements in conjunction with Computerized Adaptive Testing (CAT) and Item Response Theory (IRT).

The research follows a **two-phase approach**:

i. **Phase 1: Exploratory Design**

   Exploratory approach is employed in the first step to test the flexibility and performance of a better CAT-IRT model. A current adaptive test engine is refined in this step by collecting a dataset of labeled test questions with mappings to IRT values for difficulty, discrimination, and guessing. The primary aim in this step is to verify the model's correctness and consistency in ability prediction of the student and question selection.

ii. **Phase 2: Applied Design**

   In the second step, the validated CAT-IRT model is converted to a functional web-adaptive testing platform with gamification components. System design, architecture, UI coding, adaptive test logic implementation, and gamification aspects, including leaderboards, badges, and incentives for progress, are all part of this step.

3.2.1. The Proposed Model

This section discusses the design and workflow of the proposed system, incorporating interactive game-like features along with adaptive assessment based on CAT-IRT to enhance learner interest and comprehension of the principles of programming.

i. **Theoretical Background**

The 3-parameter logistic (3PL) model, a component of Item Response Theory models, serves as the backbone of the adaptive test engine. Difficulty, or the difficulty of an item; discrimination, or the discriminative power of a question to distinguish between high- and low-ability students; and guessing, or the chance of getting a question right, are taken into account by the 3PL model.

Through the adaptive question selection mechanism, these factors allow the system to measure the ability of a learner in real time and dynamically change the difficulty of the test.

ii. **Review of model used:**

I made use of a pre-trained BOB-CAT (Bilevel Optimization-Based Computerized Adaptive Testing) model obtained from a public GitHub repository. The BOB-CAT model was originally designed to reduce test lengths while maintaining high assessment accuracy by leveraging Item Response Theory (IRT) to adaptively select questions based on a student's estimated ability. For my implementation, I fine-tuned the model to suit the specific needs of programming education, particularly for selected data structures concepts. This involved curating a domain-specific dataset consisting of multiple-choice questions related to topics such as arrays, stacks, queues, linked lists, trees, and graphs. Each question was annotated with appropriate IRT parameters difficulty, discrimination, and guessing probability to enable effective adaptive testing. Beyond dataset preparation, I modified aspects of the model to align it with my educational goals, shifting the focus from merely minimizing test length to enhancing students' conceptual understanding. This was further integrated into a gamified platform, where the adaptive testing engine powered by the fine-tuned BOB-CAT model dynamically adjusted question difficulty based on real-time student performance. This approach not only preserved the adaptive nature of the original model but also introduced an engaging and educationally effective environment for learning programming.

iii. Model Framework / Workflow

The following are the processes that entail the model framework:

a. Input and Dataset Curation

The system employs a pre-curated collection of multiple-choice coding questions linked with IRT parameters. Questions are structured by subject, complexity, and concept to provide thorough coverage of the subject matter of data structures.

b. Fine-tuning CAT-IRT Model

The model is then fine-tuned on the annotated data using machine learning libraries (in this case, PyTorch or TensorFlow). Simulated learner responses are constructed to confirm adaptive behavior. The model learns to predict student capability and choose the next most helpful question given the answers previously seen.

c. System Design and Development

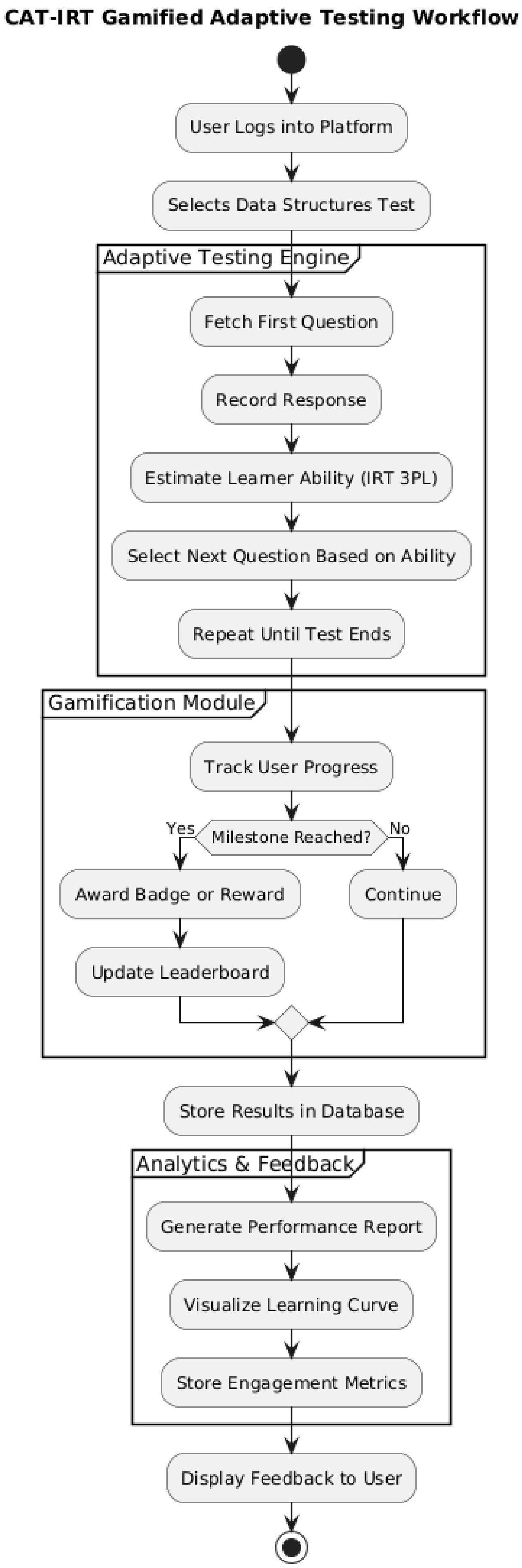
A web platform is built utilizing Next.js for the user interface and backend APIs with TypeScript. RESTful APIs link the frontend to the CAT engine to retrieve the next best question in real time.

d. Gamification Integration

The platform has badges, points and leaderboards. Badges are gotten from accomplishing milestones (e.g., completing a certain number of quizzes). Feedback is gamified to encourage learners and minimize test anxiety.

e. User Testing and Feedback Loop

Students engage with the platform through test sessions. Their answers are recorded, and performance and engagement metrics are gathered. The system applies these metrics to improve upcoming test sessions and provide a better user experience.


CAT-IRT Gamified Adaptive Testing Workflow
User Logs into Platform
Selects Data Structures Test
Adaptive Testing Engine
Fetch First Question
Record Response
Estimate Learner Ability (IRT 3PL)
Select Next Question Based on Ability
Repeat Until Test Ends
Gamification Module
Track User Progress
Milestone Reached?
Yes
No
Award Badge or Reward
Continue
Update Leaderboard
Store Results in Database
Analytics & Feedback
Generate Performance Report
Visualize Learning Curve
Store Engagement Metrics
Display Feedback to User

3.3. Curation of a dataset of data structure questions

To curate a dataset of data structure questions, I sourced standardized multiple-choice questions online. I made use of a Python web scraping script to automate the question fetching process, ensuring a broad collection of questions across key data structure topics, Trees, Graphs, Linked List, Arrays, Stack, Queue. I was able to collate 300 multichoice questions. Each question included four answer options and a designated correct answer field, each question was annotated with essential Item Response Theory (IRT) parameters, difficulty, discrimination, and guessing probability. These parameters were carefully assigned based on question complexity and the likelihood of a correct guess, using AI. After building this annotated dataset, I proceeded to simulate student interactions to generate a realistic training set for the adaptive testing model. A Python script was used to simulate responses from 500 virtual students, each attempting 30 randomly selected questions from the pool. The simulation was governed by the IRT parameters, allowing the correctness of each response to be probabilistically determined. The resulting interaction data, stored in CSV format, captured key details such as student ID, question ID, correctness of response, and associated IRT values. This dataset formed the foundation for training the adaptive model, providing it with realistic patterns of student performance in the context of data structure learning.

3.4. Design of a Database for the Gamified Adaptive Testing platform

My goal when developing the database for the gamified adaptive testing platform was to build a scalable and organized system that could accommodate the several adaptive testing components. I created a number of Mongoose schemas, each of which represents a distinct feature of the platform: User, Badge, and Courses.

The Badge schema enabled gamification by monitoring the badges obtained through user accomplishments, while the User schema oversaw user login and profile information. The TestSession property of the User schema collected data for every adaptive testing session, connecting users to the questions they were given. To guarantee flexibility and ease of maintenance, the backend was created using TypeScript and Node.js while adhering to RESTful design principles. The CAT-IRT engine, which operated as a web service or local module, was contacted by each endpoint. It used real-time estimates of the learner's ability to determine the next question.

A MongoDB database was used to safely store all user information, test sessions, answers, and performance reviews. In addition to supporting the adaptive testing procedure, the database design improved features like leaderboard rankings, performance monitoring, and individualized learning feedback, making learning more interesting and data-driven.

3.5. Fine-Tuning the Pre-existing CAT-IRT Model for Adaptive Testing

In fine-tuning the pre-existing CAT-IRT (BOBCAT) model for adaptive testing, I focused on re-training it to better evaluate students' understanding of programming concepts, with particular attention to data structures. The goal was to fine-tune the model in such a way that it could deliver a truly personalized assessment experience, dynamically selecting questions based on each learner's ability level. I started with an open-source implementation of a CAT-IRT model (BOBCAT) and fine-tuned it using the curated dataset of annotated multiple-choice questions, each tagged with IRT parameters such as difficulty, discrimination, and guessing probability. I used a Python script to standardize and organize the dataset properly to prepare it for model training. I made use of numerical scales to transform categorical values such as "difficulty" and "discrimination." To make the model even more realistic, I trained it using a simulated interaction dataset in which 500 mock students attempted 30 questions each. These interactions, generated probabilistically based on the IRT parameters, allowed the model to learn patterns in how students with varying skill levels respond to different questions. This fine-tuning process made the model more exposed to subtle differences in learner performance and allows it to select the most informative next question with greater accuracy. This fine-tuning was essential to turning the model from a general adaptive engine into one designed specially to support students in effectively and engagingly strengthening their understanding of data structure concepts.

3.6. Development of the Interface of the System

The goal is to design and create an adaptive web-based test platform fully compatible with the optimized CAT-IRT model. The platform would be the central interface through which learners would take adaptive tests.

3.6.1. System Requirements

The details of what the system will do, and what the system will have, are listed below:

i. **Functional Requirements:** The platform ought to have core adaptive testing functionality, such as user authentication, test session initialization, CAT logic-driven question delivery, and response tracking. It should support learners taking a custom sequence of questions whose difficulty level changes accordingly based on their performance. The system should expose APIs to interface with the CAT-IRT engine to select questions and estimate ability.

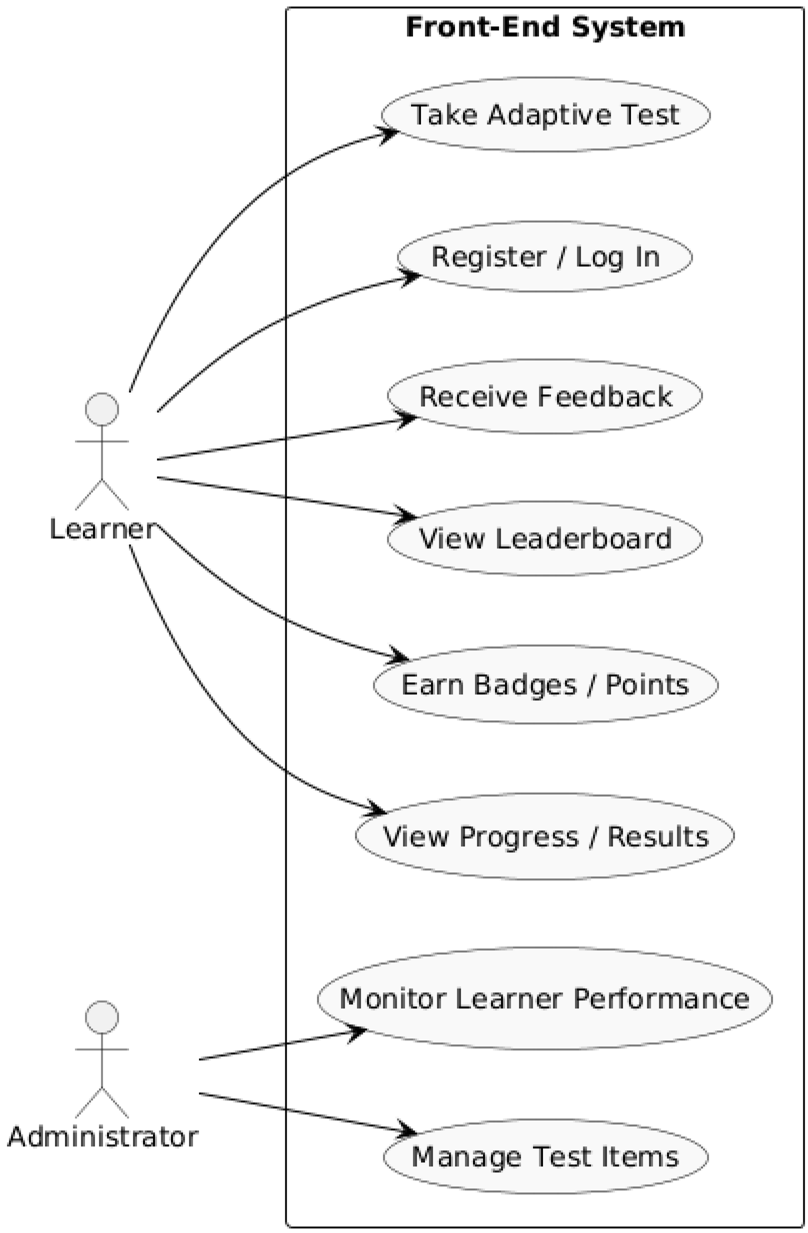
ii. **Non-functional Requirements:** The platform should be scalable to accommodate many users at once, responsive to provide real-time interaction, and secure in handling student information and test results. It should maintain low latency during the time between submitting an answer and the presentation of the next question to keep the adaptive experience intact. The UI should also be intuitive and interactive to guarantee user satisfaction and simplicity in navigation.

3.7. Implementation and Integration of Adaptive Testing in the Platform

Bringing the adaptive testing model into the actual platform was a crucial step in this study. At this stage, the goal was to create a learning experience for the users by allowing them directly interact with the CAT-IRT engine directly, users answering questions that matched their ability and getting real time response from the engine. This meant carefully embedding the adaptive engine within the core of the platform in a way that felt seamless and responsive.

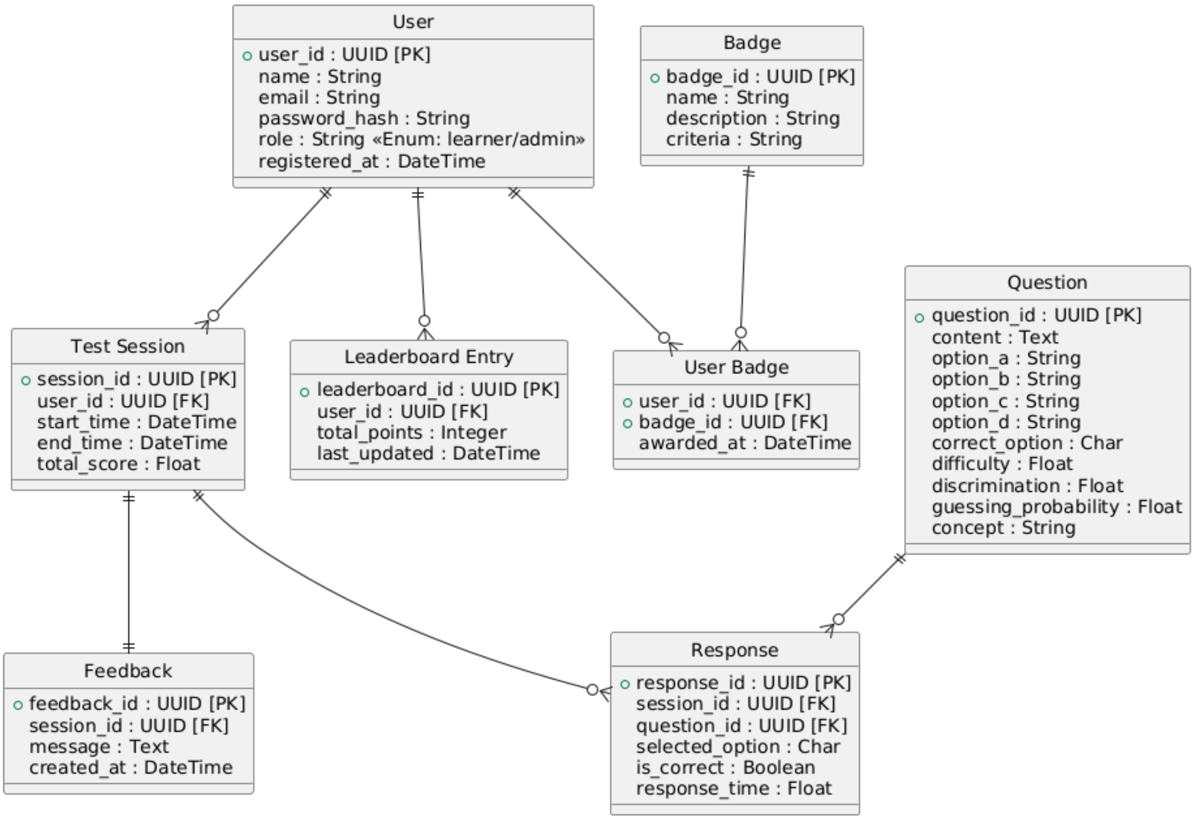
3.7.1. Front-end Implementation

The user interface was built using the Next.js framework. I gave focus to a clean and distraction-free test interface, adaptive navigation, progress tracking, and dynamically loaded questions based upon the response from the APIs. Principles of accessibility and responsive design were utilized to support multi-browser, multi-platform compatibility.



3.7.2. Back-end Implementation

The backend was developed using TypeScript and Node.js in accordance with RESTful design. Each of the endpoints talks to the CAT-IRT engine running as a web service or local module, calling functions to choose the next question based on real-time estimation of the learner's ability. User and test information are stored in a secure database, and response histories are recorded for later analysis.



3.7.3. Integration of the CAT-IRT Model to the Platform

The CAT-IRT capability is built into the platform as a central feature. The system updates and retrieves the learner's measured ability after every question via API calls and provides the next question accordingly. This enables the test to be adaptive in real-time, providing each learner with a personalized test experience. Logs and analytics are retained for real-time tracking and for performance evaluation in the future.

3.8. Evaluation of the Proposed Model using Evaluation Metrics

The evaluation of the adaptive CAT-IRT model focused on assessing its accuracy and effectiveness in predicting learner responses and selecting appropriate test items based on learner ability. The study employed a quantitative evaluation approach, analyzing model output and prediction accuracy through system-level metrics.

3.8.1. Evaluation Metrics

The following metrics were used to evaluate model performance:

i. **Accuracy:** Measures the proportion of correctly predicted learner responses.

ii. **Mean Reciprocal Rank (MRR):** Evaluates how well the model ranks questions according to learner ability.

iii. **Normalized Discounted Cumulative Gain (nDCG):** Assesses the relevance of question difficulty ordering in the adaptive testing context.

iv. **Root Mean Square Error (RMSE):** Quantifies the difference between predicted and actual learner response probabilities or ability estimates.

3.8.2. Evaluation Tools and Techniques

Python was used for all data processing and analysis tasks. Libraries such as Pandas and NumPy facilitated data manipulation, while Matplotlib and Seaborn were used for visualization. Custom scripts calculated evaluation metrics to ensure accurate performance assessment of the CAT-IRT model.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND EVALUATION

4.1. Preamble

This chapter presents an elaborate description of the approach employed to design the adaptive gamified test platform based on the CAT-IRT model developed. It also highlights the evaluation metrics used to measure the performance of the model, and the results of the evaluation.

4.2. System Requirements

This section explains the conditions in which the system was developed and expected to operate smoothly. It comprises all the software and hardware necessary to install and execute the adaptive test platform with ease.

4.2.1. Hardware Requirements

Table 4.1 below shows the major hardware requirements for the implementation of the system.

Table 4.1: Hardware Requirements

S/N	Requirement	Hardware
1.	Processor	Intel core i5 or higher/ Apple M1 chip
2.	Primary Memory	16 GB RAM or higher
3.	Architecture	64Bit (X64) for windows / ARM X64
4.	Secondary Storage	32GB HDD or higher

4.2.2. Software Requirements

A number of software pieces must be used to design, implement, and execute the adaptive gamified test system. The system takes advantage of a mix of client-side, server-side, and model-related technology to deliver a smart and interactive test experience. The primary software requirements and their respective functions are documented in Table 4.2.

Table 4.2: Software Requirements

S/N	Requirement	Software
1.	Operating System	Mac OS: v11.0.0 or higher Windows: 10 or higher Linux: Ubuntu
2.	Programming Language	Python 3, TypeScript
3.	Database Management System	MongoDB and Mongoose
4.	Development Tool	VS Code, PyCharm
5.	Frameworks	Next.js (web platform), PyTorch/TensorFlow (model training)
6.	Model Deployment	GitHub and Render
7.	Libraries	Pandas, NumPy, torch, SciPy, Flask, PyMongo, Gunicorn.

4.3. Implementation Tools

I made use of the following tools to build the adaptive testing platform. Their support for both frontend and backend, along with system architecture compliance, were the key factors to consider while selecting them. The following lists the most significant ones (the entire list can be seen in Table 4.2):

i. **Python 3:** Python was utilized to develop and refine the models and create the backend functionality of the adaptive testing model. It was easy to test and implement because of the large collection of machine learning and data science packages in its ecosystem.

ii. **TypeScript:** I made use of TypeScript for the development of the frontend and NodeJS backend. Generally, the web app was more reliable and easier to debug as a result of its static typing.

iii. **Next.js:** My go-to framework for developing the user interface of the adaptive test platform was Next.js. It performed exceptionally and supported server-side rendering, both of which were useful for responsive interfaces.

iv. **Node.js:** The backend of the online platform was powered by Node.js. It handled authentication and session handling, connection to the adaptive testing model backend, and the incorporation of gamification components such as badges, points and leader board.

v. **MongoDB:** MongoDB served as the database for the platform. It was used during model tests and evaluations to store test results, learner performance and progress information, and test logs.

vi. **Visual Studio Code (VS Code):** Visual Studio Code was my primary Integrated Development Environment (IDE) for front-end development. It was very light-weight and useful to debug code with when developing the frontend interface using TypeScript and Next.js.

vii. **PyCharm:** I made use of the PyCharm Integrated Development Environment (IDE) during the training of the model and the development of the model's backend. PyCharm is great at handling virtual environments, debugging, and large-scale codebases.

viii. **PyTorch:** I utilized PyTorch for the adaptive test model development and model optimization. PyTorch's ability to process graphs dynamically and flexibility proved useful for experimenting with the model.

ix. **TensorFlow:** TensorFlow provided efficient optimization features and improved scalability.

x. **GitHub:** GitHub was employed to control versions and collaborate collaboratively. It played a significant role in handling backups, updates, and tracking the repositories used in the development of this project.

xi. **Render:** The model and platform backend were deployed via Render. It simplified deployment without requiring a complicated infrastructure setup.

xii. **Libraries:** The creation of the API called for the use of a number of Python packages including: NumPy, PyTorch, SciPy, Flask, PyMongo and Gunicorn.

4.4. Development Methodology

The approach to development employed in this project was the Iterative and Incremental Development (IID) model. This was the best model to employ since it enabled me to develop the adaptive testing system in small functional pieces, each iteration adding new improvements, solving past problems, and addressing feedback.

I started the project with a top-down design of both the test model and web platform. As I went along, I made continual refinement on the different components particularly during the integration of the CAT-IRT model, adaptive logic, and gamification features. There were a few places where I had to redo question selection logic, user flow experiences, and the handling of model predictions based on live test results.

For instance, while working on the gamification module, I had to tweak the assignment of points and badges based on the difficulty of the test taken. Likewise, while refining the model, I iteratively experimented with various parameter configurations and handling of the data until it was optimal.

The IID approach served the project well because it allowed me to test and refine every iteration of the system in a rapid manner. It also enabled easier isolation of bugs, performance tuning, and scaling of features incrementally while keeping the platform functional and testable at each point.

4.5. Evaluation of the Adaptive Testing Model

The adaptive test model proposed was tested to see how effective it was in the selection of question difficulty and estimation of learner ability. The process of testing involved making use of common ranking and classification metrics that are used in information retrieval and adaptive learning situations.

The performance of the model was measured using the following metrics:

i. **Accuracy:** It indicates the rate at which the model predicted the learner's response accurately.
ii. **Mean Reciprocal Rank (MRR):** Assess how soon in the ranking the best fitting question (by ability match) occurs.

iii. **Normalized Discounted Cumulative Gain (nDCG):** It evaluates the performance of the model in ranking questions by relevance (difficulty matching with the ability of the student).

iv. **Root Mean Square Error (RMSE):** Represents the error in learner ability or response probability prediction.

These were calculated after the model was tested on simulated student interaction data and actual question parameters derived from Item Response Theory (IRT).

4.5.1. Evaluation Results and Discussions

Table 4.3 below shows the evaluation results of the fine-tuned adaptive testing model:

Table 4.3: Results of Model Evaluation

S/N	Metric Test	Score
1	Accuracy	0.6645
2	MRR	0.1000
3	nDCG	1.000
4	RMSE	1.3275

The following describes the evaluation metrics based on the model evaluation results shown in Figure 4.1:

i. **Accuracy (0.6645):** This indicates the model accurately predicted the learner would answer a question right in about 66.5% of the test cases. This indicates a level of moderate predictive accuracy, where the system can distinguish overall between right and wrong answers but has potential to do better.

ii. **Mean Reciprocal Rank (MRR: 0.1000):** A relatively low MRR score indicates the model tends to choose questions with the best fit later in the sequence over earlier. Though the system is picking the right questions, it might better rank and prioritize the most apposite items initially.

iii. **Normalized Discounted Cumulative Gain (nDCG: 1.0000):** A perfect score on the right represents the model doing a great job at ranking questions by relevance to the learner's assumed ability. This verifies the finding that even when the most favourable questions are perhaps not available first (since MRR would indicate), the overall ordering is optimally placed in terms of difficulty matching.

iv. **Ability RMSE (1.3275):** This indicates the overall error in estimating learner capability. A figure of ~1.33 indicates the system to be estimating capability in the right ballpark, but with some deviation between predicted and observed values. This could be enhanced with further fine-tuning of the IRT parameters or adaptive choice logic.

v. **Probability RMSE (0.2140):** This metric gauge the predictive accuracy in terms of the error in response probabilities. A measure of 0.2140 reflects good predictive precision, which indicates the model's confidence in a learner answering a question correctly or incorrectly being reasonably accurate.

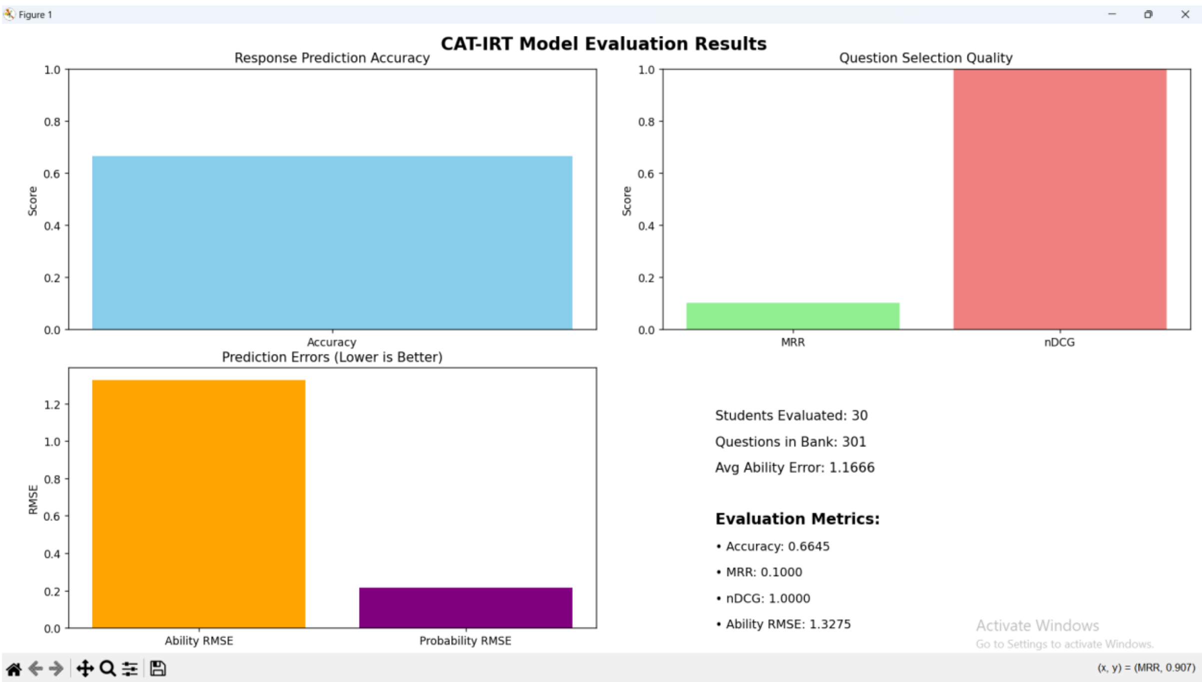


Figure 4.1: Model Evaluation Results

4.6. Program Modules and Interfaces

Several modules and user interfaces were implemented in this project. They include the login/signup interfaces, the course list and topics list interface, the active test interface, and the profile interface.

4.6.1. Login/Signup Interface

This page is where the user gets to sign up and sign into the system using their email and password.

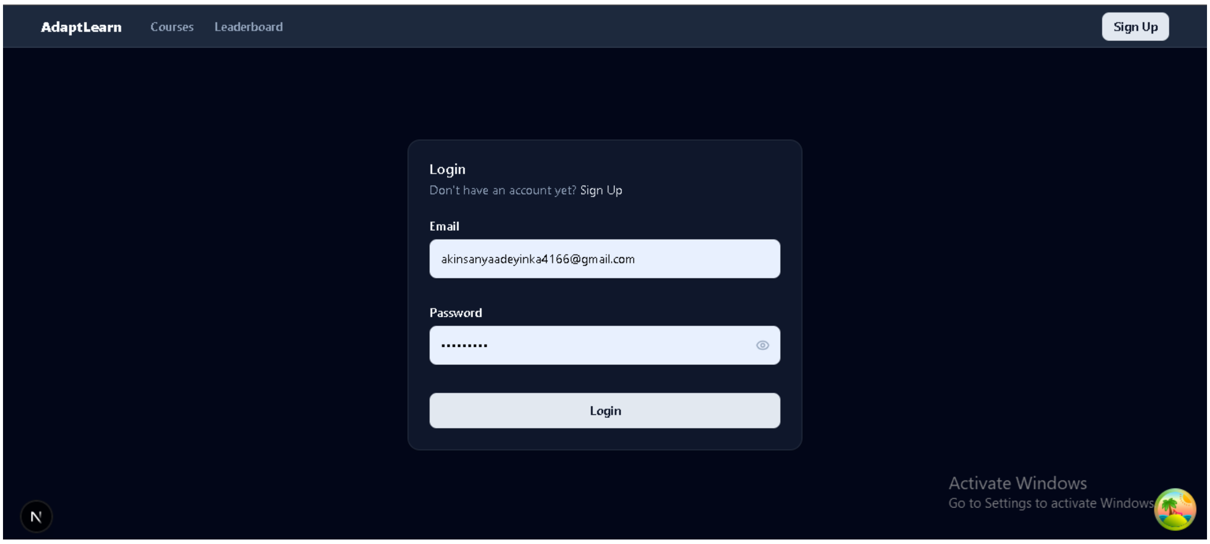


Figure 4.2: Login Interface

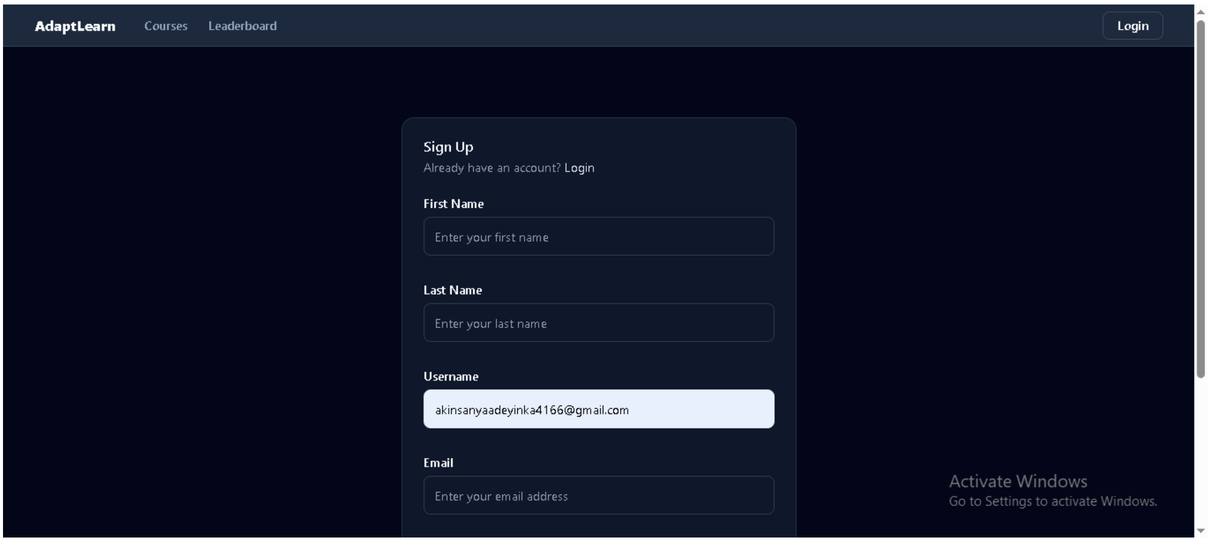


Figure 4.3: Signup Interface

4.6.2. The Web Application Features

These pages are where the user can interact with the web application by viewing topics, taking quizzes, earning points and badges, viewing leader board, etc.

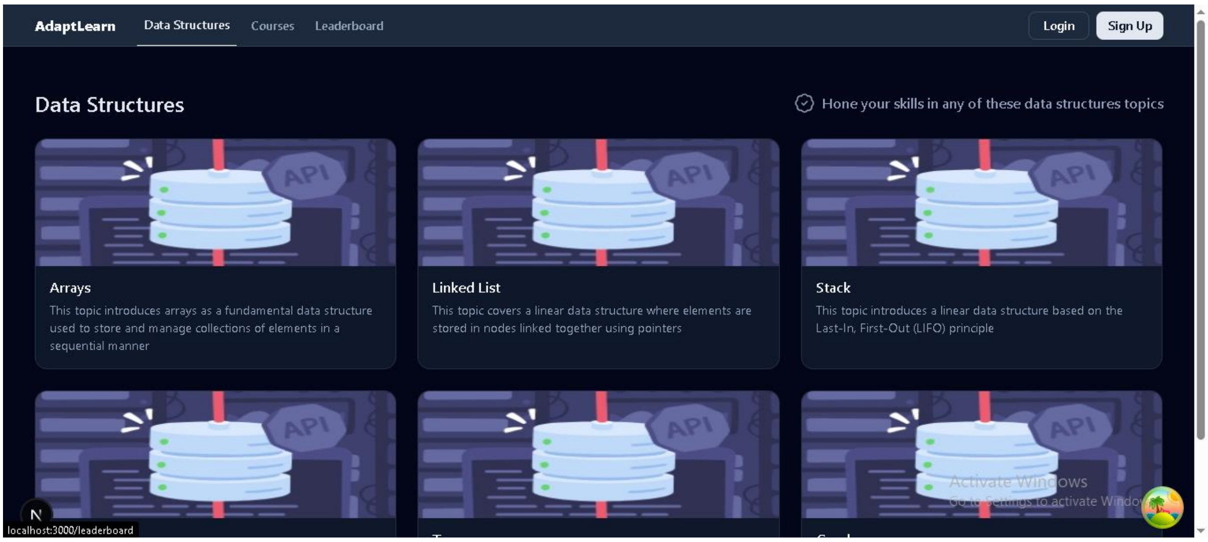


Figure 4.4: Topics List Interface

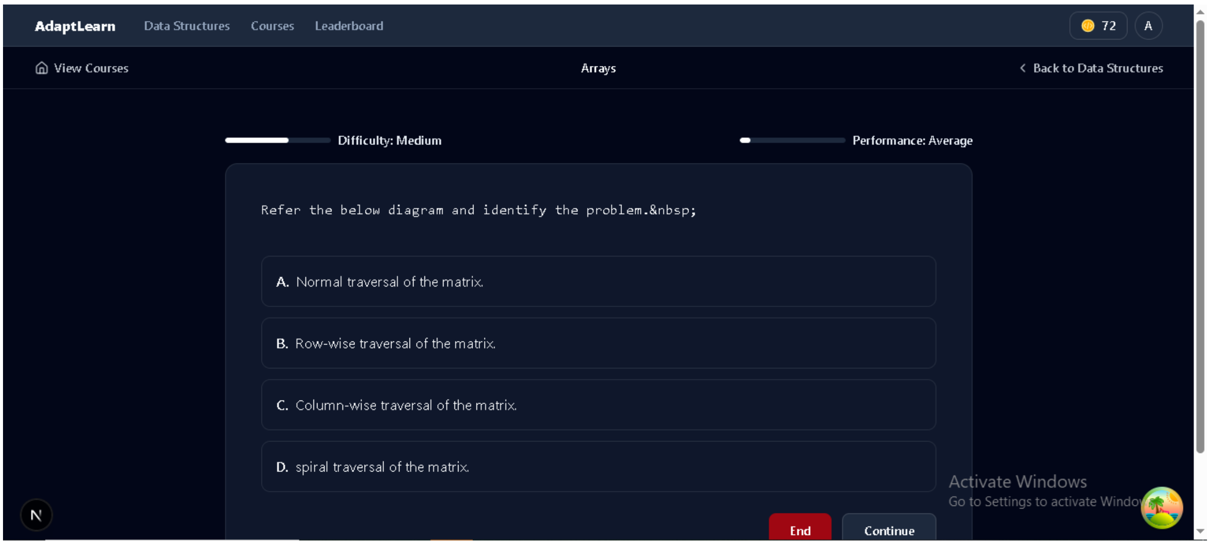


Figure 4.5: Active Test Interface

4.7. Exploratory Data Analysis (EDA) of the Dataset

Exploratory Data Analysis (EDA) was conducted on the simulated interaction data to understand the properties and structure of the data used for the training and evaluation of the CAT-IRT based adaptive test model.

4.7.1. Dataset Overview

Responses from simulated learners to a sequence of multiple-choice questions with IRT parameters form the dataset. Each row representing an independent learner-question interaction includes the following fields: Student ID, Question ID, Correct (0 or 1), Difficulty (continuous scale, typically 0 to 1), Discrimination (continuous scale), Guessing Probability (continuous or categorical), and (categorical topic tag).

4.7.2. Key Observations

The following are some key observations gotten from the exploratory data analysis:

i. **Question Distribution:** With question IDs from 1 to 301, the pool comprises 301 different questions. These questions represent a substantial question bank for adaptive testing since the student answers are scattered among them.

ii. **Student Coverage:** There are a total of 15,000 interaction records with 500 unique students (Student IDs 0 to 499) answering a series of questions. This gives a good amount of student-question interaction data for model training.

iii. **Difficulty Levels:** With a roughly 1.5 mean, the difficulty values range from about 0.5 to 2.5. The fairly evenly distributed difficulty allows for adaptive testing across easy to hard questions.

iv. **Discrimination Values:** To discriminate among various levels of learners, discrimination values also vary between 0.5 and 2.5 roughly, showing a range of questions with low to high discrimination.

v. **Correct Response Ratio:** The correct response ratio is the proportion of total responses that were answered correctly in the dataset. Figure 4.6 shows the distribution of responses. The value was calculated as:

$$\text{Correct response ratio} = \frac{Number\ of\ Correct\ Responses}{Total\ Responses} = \frac{5600}{15000} = \approx 0.375$$

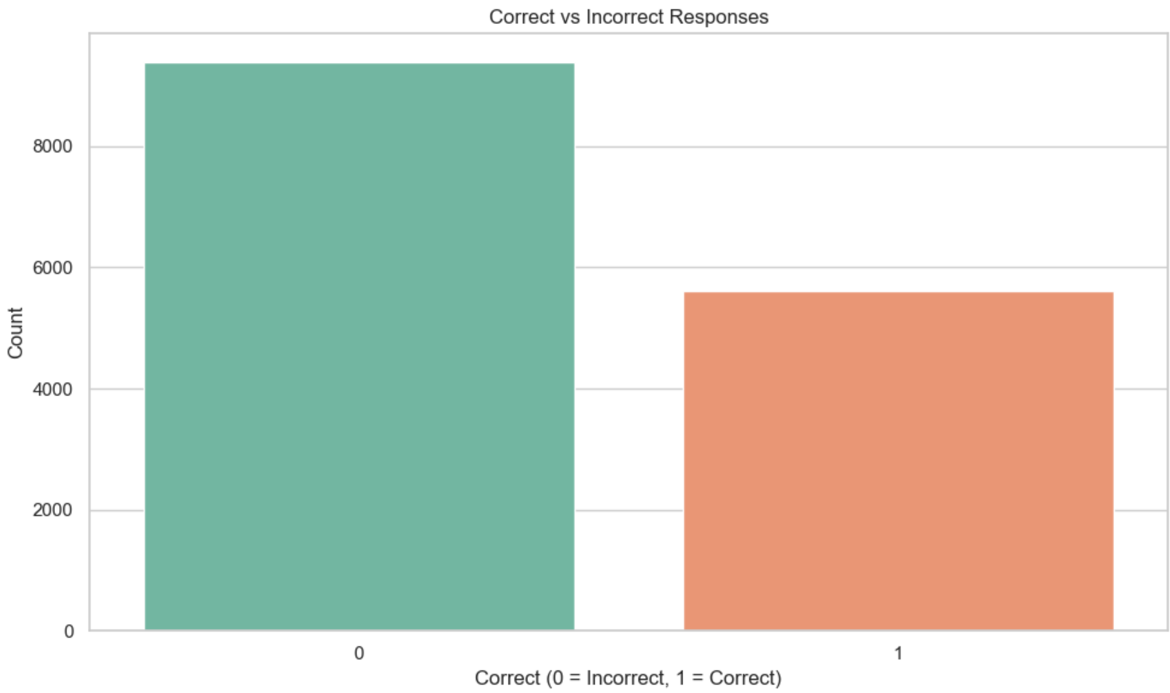


Figure 4.6: Response Correctness Distribution

vi. **Guessing Probability Parameter:** A realistic low probability of guessing consistent with standard 3PL IRT assumptions is embodied in the guessing probabilities, which are between 0.20 and 0.30 with a mean of approximately 0.25. Figure 4.7 shows the distribution of guessing probabilities.

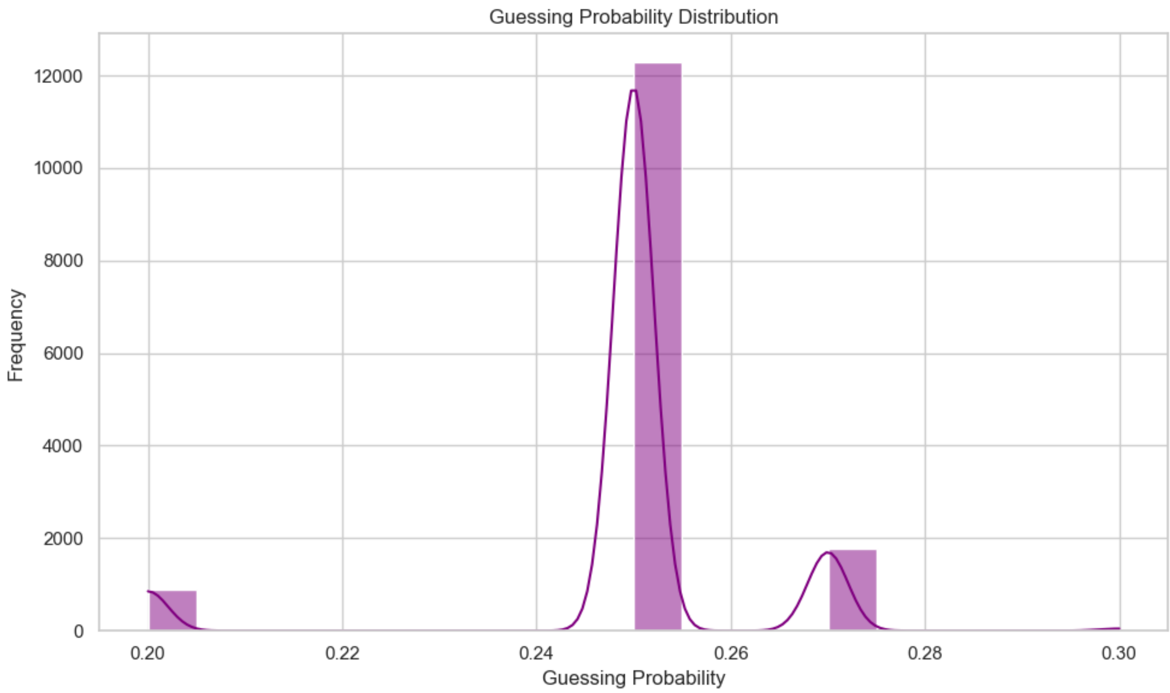


Figure 4.7: Guessing Probability Distribution

vii. **Concept Coverage:** Figure 4.8 shows the six distinct concepts (arrays, linked lists, stacks, queues, trees, and graphs) that are available at various frequencies throughout the dataset, allowing measurement of student performance across a range of topics.

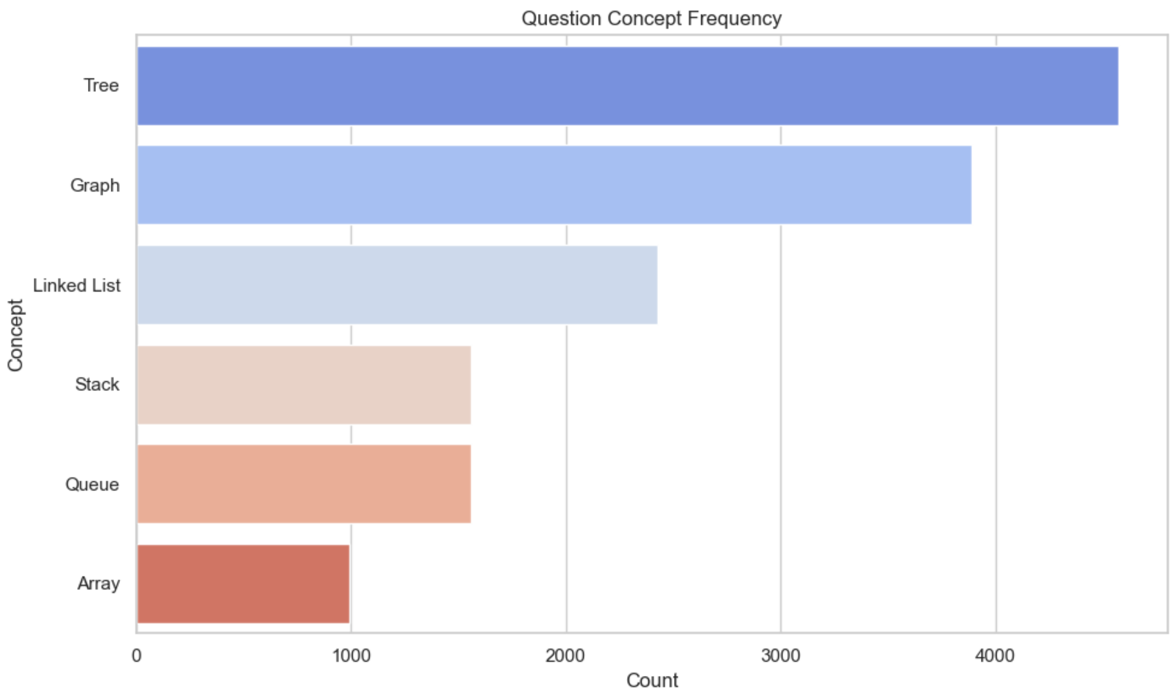


Figure 4.8: Concept Frequency Distribution

CHAPTER 5

SUMMARY, RECOMMENDATIONS, AND CONCLUSION

5.1. Summary

The idea behind this project was born out of the issues normally posed to learners in the field of programming education, particularly in the acquisition of the elementary concepts of data structures. Conventional evaluation tools make use of static tests that are not responsive to the learner’s skill level, hence giving questions either too hard or too simple, which prevents effective learning as well as motivation. The absence of interactive assessing conditions lowers learner motivation and usually results in the inadequate retention of the concepts in programming.

This project aimed to create a novel adaptive test system improving learners' knowledge of programming through the application of Computerized Adaptive Testing (CAT) founded on Item Response Theory (IRT) and gamification features. The overall purpose was to maximize test design for the concept of data structure to create personalized tests dynamically altering question difficulty according to learners' performance in real-time. This will create an effective learning path optimized for each student's requirements.

The study commenced with a thorough survey of adaptive testing methods and measurement models, and was next followed by the development of a CAT-IRT framework to estimate the ability of a learner and to choose questions in a way that maximizes accuracy. To mimic real educational conditions, a student interaction dataset was simulated, covering answers to different questions over a range of data structure topics, including linked lists, stacks, queues, trees, and graphs. This dataset served as a basis for model adaptation and evaluation.

Also, aspects of gamification were added to both the front-end and backend elements to further encourage increased user involvement. These include interactive user interfaces, tracking of progress, rewards in form of points and badges, and motivational loops to keep the learners motivated and to promote ongoing participation.

The system was extensively tested using standard performance metrics of Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (nDCG), accuracy, and Root Mean Square Error (RMSE). The metrics showed that the CAT-IRT model is effective in accommodating various levels of learner abilities through the selection of questions having proper difficulty and discrimination values, thus increasing ability estimation accuracy. The gamified context further assisted by increasing the motivation and interaction time of learners.

5.2. Recommendations

On the basis of the results and noted limitations of this project, a number of suggestions can be made to strengthen and extend the system's functionality:

i. **Dataset Enlargement and Variety:** To enhance the precision and applicability of the CAT-IRT model further, larger and diversified datasets should be incorporated in the future. Adding coverage of a wider variety of programming themes with the inclusion of themes outside the domain of data structures, for instance, algorithms, design patterns, and software development processes, would allow the system to support broader curricula and learners.

ii. **Mobile and Cross-Platform Compatibility:** Having a fully adaptive mobile version will enhance accessibility, enabling learners to take advantage of the adaptive test system at any time and from any place.

iii. **Integration with Learning Management System (LMS):** Integrating the adaptive test platform with leading LMS platforms would enable straightforward deployment to educational institutions and the easier monitoring of learners' progress.

iv. **Extension to Collaborative Learning Modes:** Looking into multi-user test modes in which learners are allowed to participate in peer challenges or group problem-solving would tap social learning advantages.

5.3. Conclusion

This final project successfully designed and developed a gamified Computerized Adaptive Testing system based on Item Response Theory to implement tests with the potential to advance learners' understanding of data structures. The system proved to possess the effective capability to customize tests dynamically choosing questions on the basis of the learners' respective abilities, thereby overcoming the shortcomings of conventional static tests.

The incorporation of gameplay elements across the platform increased learner motivation and engagement, making the assessment process both interactive and fun. Verifications with key metrics of MRR, nDCG, accuracy, and RMSE established that the CAT-IRT model offered valid and accurate ability estimation, while the gamification elements ensured that learners remain engaged.

This project pushes computational education forward by combining state-of-the-art psychometric design with motivational game design to enable personalized pathways of learning. Though further refinement of model accuracy and coverage of subject material may be forthcoming, the project provides a firm basis for ongoing development of intelligent, adaptable, and engaging assessment tools for computer programming.

Aside from its scholarly contribution, the system has considerable practical application in education, providing instructors with a scalable and data-driven approach to test and improve learners' comprehension at an efficient rate. This work underscores the possibility of using adaptive testing and gamified learning to revolutionize education assessment and enhance performance of students in computer science and allied disciplines.

REFERENCES

Boguslawski, S., Deer, R. & Dawson, M. G. (2024). Programming education and learner motivation in the age of generative AI: Student and educator perspectives. *Information and Learning Science*. https://doi.org/10.1108/ILS-10-2023-0163

Ebenbeck, N. (n.d.). *Computerized Adaptive Testing in Inclusive Education*. https://doi.org/10.5283/epub.54551

Hasan, M. N. (2024). *Introduction to Data Structures Data Structures and Algorithms Course Code :CCE-2301*. https://www.researchgate.net/publication/384707448_Introduction_to_Data_Structures_and_Algorithms

Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M. & Lim, H. (2020). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior*, *114*, 106575. https://doi.org/10.1016/j.chb.2020.106575

Jaramillo-Mediavilla, L., Basantes-Andrade, A., Cabezas-González, M., & Casillas-Martín, S. (2024). Impact of gamification on motivation and academic performance: A systematic review. *Education Sciences*, *14*(6), 639. https://doi.org/10.3390/educsci14060639

Karsen, M., Masrek, M. N., Uzir, N. A. & Safawi, A. R. (2022). Gamification in MOOC: A systematic literature review. *Environment-Behaviour Proceedings Journal*. https://doi.org/10.21834/ebpj.v7isi10.4111

Latu, E., & Chapman, E. (2002). Computerised adaptive testing. *British Journal of Educational Technology*, *33*(5), 619–622. https://doi.org/10.1111/1467-8535.00296

Lavoué, E., Ju, Q., Hallifax, S. & Serna, A. (2021). Analyzing the relationships between learners' motivation and observable engaged behaviors in a gamified learning environment. *International Journal of Human-Computer Studies*, *154*, 102670. https://doi.org/10.1016/j.ijhcs.2021.102670

Li, C., Liang, L., Fryer, L. K. & Shum, A. (2024). The use of leaderboards in education: A systematic review of empirical evidence in higher education. *Journal of Computer Assisted Learning*. https://doi.org/10.1111/jcal.13077

Ma, H., Zeng, Y., Yang, S., Qin, C., Zhang, X. & Zhang, L. (2023). A novel computerized adaptive testing framework with decoupled learning selector. *Complex and Intelligent Systems*, *9*(5), 5555–5566. https://doi.org/10.1007/s40747-023-01019-1

Ogunsakin, I. B. & Shogbesan, Y. (2018). *Item Response Theory (IRT): A Modern Statistical Theory for Solving Measurement Problem in 21st Century*. http://www.ijsre.com.

Oliveira, W., Hamari, J., Shi, L., Toda, A. M., Rodrigues, L., Palomino, P. T. & Isotani, S. (2022). Tailored gamification in education: A literature review and future agenda. *Education and Information Technologies*, *28*(1), 373–406. https://doi.org/10.1007/s10639-022-11122-4

Papanastasiou, E. C. (2003). *Computer-Adaptive Testing in Science Education*. https://www.researchgate.net/publication/251619785

Rizzardini, R. H., Chan, M. M. & Guetl, C. (2016). An Attrition Model for MOOCs. *Elsevier eBooks*. https://doi.org/10.1016/b978-0-12-803637-2.00014-2

Ružic, I. M., & Dumancic, M. (2015). Gamification in education. *Informatologia*, *48*(3–4), 198–204. https://doi.org/10.3390/encyclopedia3040089

Triantafyllou, S. A., Sapounidis, T., & Farhaoui, Y. (2024). Gamification and computational thinking in education: A systematic literature review. *Salud, Ciencia y Tecnología – Serie de Conferencias, 3*(659). https://doi.org/10.56294/sctconf2024659

Troussas, C., Krouska, A. & Sgouropoulou, C. (2019). Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education. *Computers & Education*, *144*, 103698. https://doi.org/10.1016/j.compedu.2019.103698

Weiss, D. J. & Kingsbury, G. G. (1984). Application of computerized adaptive testing to educational problems. *Journal of Educational Measurement*, *21*(4), 361–375. https://doi.org/10.1111/j.1745-3984.1984.tb01040.x

Wiredu, J. K., Abuba, N. S. & Acheampong, R. W. (2024). Enhancing accessibility and engagement in computer science education for diverse learners. *Asian Journal of Research in Computer Science*. https://doi.org/10.9734/ajrcos/2024/v17i10509

Zhang, Z. & Huang, X. (2024). Exploring the impact of the adaptive gamified assessment on learners in blended learning. *Education and Information Technologies*. https://doi.org/10.1007/s10639-024-12708-w

Zourmpakis, A. I., Kalogiannakis, M. & Papadakis, S. (2024). The effects of adaptive gamification in science learning: A comparison between traditional inquiry-based learning and gender differences. *Computers*, *13*(12). https://doi.org/10.3390/computers13120324