PREPMASTER: AN ADAPTIVE MCQ BASED SKILL EVOLUTION AND PROGRESSION SYSTEM

Senevirathna D.M.O.C.

IT21286650

BSc (Hons) Degree in Information Technology

Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

PREPMASTER: AN ADAPTIVE MCQ BASED SKILL EVOLUTION AND PROGRESSION SYSTEM

Senevirathna D.M.O.C.

IT21286650

Dissertation submitted in partial fulfilment of the requirements for the Bachelor of Science (Hons) in Information Technology Specialized in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Group Member Name	Student ID	Signature
Senevirathna D.M.O.C.	IT21286650	

Signature:

Date: 10/04/2025

The supervisor should certify the dissertation with the following declaration.

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Date: 11/04/2025

Signature of the supervisor:

Sri Lanka Institute of Information Technology

ABSTRACT

This research project presents an innovative adaptive learning platform called "PrepMaster: MCQ Level-Up System," specifically developed to improve the interview preparedness of Software Engineers, Quality Assurance Engineers, and Product Managers. The MCQ Level-Up System dynamically evaluates and enhances user skills through personalized, progressively challenging multiple-choice questions (MCQs). Advanced adaptive methodologies and machine learning models, including Deep Q-Network (DQN), are integrated to ensure precision in skill estimation and optimal user progression.

The theoretical framework combines Item Response Theory (IRT) and Vygotsky's Zone of Proximal Development (ZPD). IRT quantitatively assesses user skill levels and question difficulty, providing statistically grounded assessments that guide adaptive question selection. ZPD ensures tasks presented are slightly beyond current user skill levels, fostering continuous growth without overwhelming users.

The MCQ Level-Up System's architecture includes several key components: a user interaction interface allowing users to select their role (SE, QA, PM), initial skill assessments for baseline proficiency determination, and an IRT skill estimation module to dynamically measure proficiency. Users are categorized into Beginner, Intermediate, or Expert levels based on their skill estimates. The DQN model then selects appropriate questions within the user's ZPD, maximizing engagement and learning outcomes. Real-time feedback mechanisms update skill estimation and refine the DQN model continually.

Performance metrics such as accuracy rates, progression speed, user retention, and IRT estimation accuracy (RMSE) validate the system's effectiveness. Data collection includes a comprehensive question dataset categorized by role, difficulty, and cognitive complexity, alongside detailed user response datasets. Future enhancements will expand question diversity, refine algorithms, integrate voice interactions, and enhance real-time adaptive feedback, further strengthening interview preparedness.

Key Words: - Adaptive Learning, Personalized Skill Evaluation, Reinforcement Learning, Deep Q-Network (DQN), MCQ Level-Up System

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, whose continuous guidance, expertise, and encouragement have been invaluable throughout this research. Their insightful feedback and unwavering support have played a crucial role in shaping this project, helping me overcome challenges and refine my work to achieve meaningful results. Their dedication to mentoring and knowledge-sharing has been truly inspiring, and I am incredibly fortunate to have had their guidance during this academic journey.

I extend my sincere appreciation to my team members for their collaboration, dedication, and hard work. The success of this project would not have been possible without their contributions, creative ideas, and willingness to work together to solve complex problems. Their commitment to achieving our common goals made the research process both productive and enjoyable. I am grateful for the teamwork and the valuable experiences we have shared throughout this journey.

I would also like to thank the faculty members and external mentors who provided their expertise and valuable insights. Their constructive feedback and technical guidance have significantly enriched my understanding of the subject matter and helped refine various aspects of this research. Their support and willingness to share their knowledge have been instrumental in the successful completion of this project.

Furthermore, I am deeply appreciative of my family and friends, whose unwavering support has been a source of strength and motivation. Their constant encouragement, patience, and belief in my abilities helped me stay focused and committed to my research, even during challenging times. Their emotional and moral support have played a significant role in keeping me motivated throughout this journey.

Lastly, I would like to extend my gratitude to everyone who, in any way, contributed to the completion of this research. Whether through direct assistance, words of encouragement, or providing a conducive environment for my studies, each contribution has been deeply valued. This journey has been both challenging and rewarding, and I am grateful to have had the support of so many individuals along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
DQN	Deep Q-Network
IRT	Item Response Theory
ML	Machine Learning
MCQ	Multiple Choice Question
MDP	Markov Decision Process
MySQL	My Structured Query Language
NLP	Natural Language Processing
QA	Quality Assurance
RMSE	Root Mean Square Error
RL	Reinforcement Learning
SE	Software Engineering
PM	Product Management *(or Project Management)*
TF-IDF	Term Frequency-Inverse Document Frequency
UI	User Interface
ZPD	Zone of Proximal Development

1. INTRODUCTION

In today’s competitive job market, technical proficiency alone is insufficient for career advancement, as the importance of effective interview performance continues to rise. Organizations increasingly demand candidates who not only possess deep domain-specific knowledge but can also demonstrate their expertise effectively during interviews. [1] Traditional methods of interview preparation, typically reliant on static, standardized practice tests or self-study guides, often fail to deliver personalized learning experiences, leaving candidates inadequately prepared for the adaptive and dynamic nature of modern job interviews. Acknowledging these limitations, this research introduces "PrepMaster: MCQ Level-Up System," [2] a state-of-the-art adaptive learning platform specifically designed to enhance interview preparedness for Software Engineers, Quality Assurance Engineers, and Product Managers. Unlike conventional preparation methods, PrepMaster dynamically adapts question difficulty based on real-time performance metrics, ensuring an individually tailored learning experience. This innovative approach provides learners with structured, progressive skill enhancement, significantly increasing their readiness for technical job interviews.

The theoretical underpinning of PrepMaster integrates two robust educational frameworks: Item Response Theory (IRT) and Vygotsky’s Zone of Proximal Development (ZPD). Item Response Theory quantitatively evaluates user proficiency alongside the difficulty of multiple-choice questions, thus ensuring precise alignment of question difficulty with the user's skill level. [3] Meanwhile, the Zone of Proximal Development emphasizes presenting tasks slightly beyond the learner's current capability, promoting continuous growth without triggering frustration or disengagement. To achieve adaptive and personalized learning, PrepMaster employs advanced machine learning methodologies, notably the Deep Q-Network (DQN). This reinforcement learning algorithm intelligently selects subsequent questions based on users' real-time responses, effectively balancing user skill enhancement with engagement. [4] The integration of DQN enables the system to continuously adapt and evolve, optimizing question selection for maximum educational impact and efficient skill acquisition.

From an architectural perspective, PrepMaster is built with a robust three-tier structure comprising a user-friendly frontend interface (React.js), efficient backend systems (Flask and Spring Boot), and secure data management through MySQL databases. Users interact with the system by initially selecting their professional role, undergoing baseline skill assessments, and progressing through tailored learning modules designed to accurately reflect their proficiency level. Real-time feedback mechanisms continually refine skill estimations and improve adaptive question selection, delivering a seamless and impactful learning experience.

This research seeks to bridge the existing gaps in traditional interview preparation methodologies by offering a systematic, intelligent, and responsive learning environment. The effectiveness of the PrepMaster system is validated through rigorous evaluation metrics, including accuracy rates, progression dynamics, user retention statistics, and estimation accuracy. By continuously adapting to user needs and performance levels, PrepMaster significantly enhances the preparedness of candidates, ultimately equipping them with the necessary competencies to excel in the dynamic landscape of technical interviews.

1.1 Background

In recent years, the job market has undergone significant transformations, driven by rapid technological advancements and evolving employer expectations. As companies increasingly prioritize specialized technical expertise coupled with robust problem-solving abilities, traditional methods of interview preparation have proven insufficient. Candidates frequently struggle to demonstrate their full potential due to the static nature of conventional preparation tools, which often fail to adapt to diverse skill levels and learning needs.

The limitations inherent in traditional multiple-choice question (MCQ) systems have spurred interest in adaptive educational technologies. Static MCQ frameworks typically lack personalized difficulty adjustments, leading to either insufficient challenges or overwhelming complexity, both of which can negatively affect candidate performance and morale. [5]

Researchers have emphasized the need for adaptive learning models capable of dynamically responding to user proficiency, thereby enhancing engagement and learning efficacy. Educational theories such as Item Response Theory (IRT) and the Zone of Proximal Development (ZPD) have been identified as critical methodologies to guide effective adaptive assessment systems.

To address these challenges, innovative adaptive systems leveraging machine learning, particularly reinforcement learning models like Deep Q-Networks (DQN), have emerged as promising solutions. These models dynamically select questions and tailor difficulty based on user responses, ensuring personalized and progressive learning experiences. The integration of advanced adaptive methodologies within interview preparation systems has the potential to significantly improve user engagement, knowledge retention, and overall interview performance, directly responding to the evolving demands of modern professional environments.

1.2 Literature Survey

In recent educational technology research, adaptive learning systems have been extensively explored to address limitations in traditional static learning methods. This has led to the development of personalized learning frameworks that dynamically adjust content based on user proficiency and performance. PrepMaster leverages these principles, combining adaptive learning methodologies to enhance user engagement and knowledge retention. [6]

Adaptive educational systems frequently incorporate robust theoretical frameworks to improve precision and effectiveness. Item Response Theory (IRT), provides quantitative assessment methods, evaluating user proficiency and question difficulty to ensure accurate skill alignment. Complementing this, Vygotsky's Zone of Proximal Development (ZPD) recommends tasks slightly above a learner's current capabilities to facilitate continual cognitive growth without causing frustration. PrepMaster strategically integrates both IRT and ZPD to foster effective learning through accurately tailored difficulty levels. [7]

Reinforcement learning (RL), particularly Deep Q-Networks (DQN), has become increasingly prevalent in adaptive testing applications due to its dynamic decision-making capabilities. By employing DQN, PrepMaster intelligently selects appropriate questions based on real-time user performance, continuously optimizing the learning experience and maximizing educational outcomes.

Comparative studies with existing adaptive learning platforms such as Coursera, Udacity, LeetCode, and HackerRank reveal notable limitations in role-specific assessments and dynamic difficulty adjustments. PrepMaster addresses these gaps by providing tailored role-based MCQs and real-time adaptive feedback, significantly outperforming traditional systems in user engagement and knowledge retention. [8] This approach is validated through rigorous performance metrics, positioning PrepMaster as a comprehensive solution for effective and personalized interview preparation in technical fields.

Table 1: Comparison with Existing Intelligent Learning Systems.

Features	Traditional MCQ Systems [9]	Knewton /ALEKS [10]	LeetCode /HackerRank [11]	MCQ Levelup System
Adaptive Questioning	No	Yes	Partial	Yes
Role-Specific Content	No	No	Partial	Yes
Reinforcement Learning	No	Yes	No	Yes
Real-Time Feedback	No	Yes	Partial	Yes
User Progression Tracking	Limited	Yes	Yes	Yes
Dynamic Difficulty Adjustment	No	Partial	No	Yes

1.3 Research Gap

Despite the advancements in online learning platforms and adaptive education technologies, a notable gap persists in the field of interview preparation systems, particularly those tailored for specific job roles. Existing platforms such as LeetCode, HackerRank, Coursera, and Udacity offer practice environments and basic adaptive features but lack deep personalization based on real-time performance and role-specific competency tracking. These systems often present users with predefined question sets and static progression models, which fail to consider individual learning trajectories and skill variations across different professional domains.

Most conventional MCQ-based assessment systems operate on a one-size-fits-all model, ignoring the varying cognitive loads and skill acquisition paths of users. [12] They rarely integrate psychometric principles like Item Response Theory (IRT) or educational models such as the Zone of Proximal Development (ZPD), both of which are essential for creating meaningful and scalable adaptive learning systems. As a result, users may either encounter questions that are too easy, leading to disengagement, or too difficult, causing frustration, both of which hinder effective learning and accurate skill assessment.

Another critical gap lies in the absence of intelligent progression and feedback mechanisms. Traditional systems often do not provide real-time feedback tailored to individual performance. They may lack adaptive algorithms like reinforcement learning (RL) or Deep Q-Networks (DQN), which can dynamically adjust the learning pathway. [13] Without these technologies, the systems are unable to identify and respond to changing user performance patterns, making them ineffective in providing a truly personalized and evolving learning experience.

Furthermore, there is a scarcity of platforms that focus on multi-role adaptability. While many platforms focus solely on programming or software development, few cater to other critical roles like Quality Assurance (QA) or Project/Product Management (PM). [14] This lack of role-specific content limits the system's applicability and relevance for users preparing for diverse technical interviews, ultimately reducing their preparedness for real-world job scenarios.

This research addresses these gaps by introducing the PrepMaster: MCQ Level-Up System, which combines IRT, ZPD, and reinforcement learning (DQN) to build a dynamic and role-specific adaptive learning platform. [15] It ensures structured difficulty progression, real-time feedback, and personalized skill assessment for Software Engineers, QA Engineers, and PMs. By bridging theoretical concepts with practical implementation, this system fills a significant void in the landscape of intelligent interview preparation tools and sets a new standard for adaptive learning in professional development.

1.4 Research Problem

In the realm of technical interview preparation, candidates often face the challenge of navigating a highly competitive environment where expectations extend beyond basic theoretical knowledge. While existing learning platforms provide question-based training, they typically lack the ability to assess user-specific learning patterns and adjust accordingly. This presents a significant problem: users are often presented with questions that are either not relevant to their job role or are not suited to their actual skill level, which results in inefficient learning and lower confidence during interviews.

Most traditional MCQ-based preparation tools follow a fixed-difficulty model where questions are not adapted based on real-time performance. This one-dimensional approach overlooks the learner's evolving understanding and fails to maintain an appropriate level of challenge. Users may find themselves repeatedly facing questions that are too easy or too difficult, leading to disengagement, frustration, and ultimately suboptimal preparation outcomes. The lack of personalization severely limits the system's effectiveness in nurturing actual skill progression.

Another dimension of the problem lies in the generic nature of the content provided. Many platforms focus predominantly on software development roles, neglecting other critical fields such as Quality Assurance (QA) and Project or Product Management (PM). This narrow scope prevents users from different domains from receiving contextualized, job-relevant assessments. As a result, even skilled professionals in QA or PM roles find themselves underprepared for their specific interview scenarios due to a lack of targeted practice.

Additionally, traditional MCQ systems rarely offer continuous feedback or real-time adaptation based on user interactions. [16] This results in a fragmented learning process where users are not adequately guided to identify their weaknesses or build upon their strengths. Without data-driven insights and real-time support, users struggle to understand their progress or determine the best way to improve, thereby reducing motivation and learning efficiency.

The core research problem this study addresses is the lack of a personalized, role-specific, adaptive MCQ assessment system that can dynamically respond to a user's learning progression. By integrating psychometric models like IRT, educational theories like ZPD, and advanced reinforcement learning techniques such as Deep Q-Networks (DQN), this research aims to develop an intelligent platform capable of delivering customized learning paths, real-time feedback, and measurable progression for diverse technical roles. Solving this problem is vital for enabling learners to bridge the gap between knowledge and performance, leading to improved interview outcomes and professional readiness. [17]

1.5 Research Objectives

1.5.1 Main Objective

To design and implement an adaptive MCQ-based skill assessment and progression system that leverages reinforcement learning and educational psychology to enhance interview preparedness for Software Engineers, Quality Assurance Engineers, and Product Managers.

1.5.2 Specific Objectives

- To develop a dynamic question selection mechanism using Deep Q-Network (DQN) reinforcement learning to adaptively personalize assessments based on user performance.
- To integrate Item Response Theory (IRT) and Zone of Proximal Development (ZPD) models for accurate skill estimation and progression tracking.
- To construct a role-specific MCQ question bank categorized by topic, difficulty level, and cognitive complexity (Bloom's taxonomy).
- To implement real-time feedback and performance analytics that help users identify strengths and areas for improvement.
- To evaluate the system's effectiveness in improving user engagement, accuracy, and knowledge retention compared to traditional MCQ assessments.
- To design a scalable and user-friendly web platform using React.js for the front-end and Flask & Spring Boot for the backend, hosted on cloud infrastructure.

2. Methodology

The methodology of this research plays a pivotal role in achieving the core objective of developing a personalized and adaptive learning platform tailored for technical interview preparation. Unlike traditional MCQ systems that follow a linear and static design, this research emphasizes dynamic content delivery using intelligent algorithms, real-time feedback, and role-specific assessments. A structured and iterative approach was adopted to ensure the system responds to diverse user profiles across Software Engineering, Quality Assurance, and Project Management domains.

The research employs an experimental and applied methodology to bridge the gap between conventional static assessments and modern AI-driven learning systems. This process begins with the identification and collection of relevant data sets, followed by systematic preprocessing and integration of machine learning models. Each stage of development is anchored in well-established educational theories such as Item Response Theory (IRT) and the Zone of Proximal Development (ZPD), which provide the foundational logic for adaptive learning and skill assessment.

Additionally, reinforcement learning models, particularly the Deep Q-Network (DQN), were utilized to make intelligent decisions in selecting the most appropriate questions for users. By modeling user progression as a Markov Decision Process, the system can adjust to the learner's performance in real time. The architectural design supports a scalable, cloud-based deployment that allows users to access the system from various environments, ensuring practicality and accessibility for large-scale implementation.

Overall, this methodology aims to deliver a high-impact, data-driven learning experience that simulates real-world interview scenarios. It ensures that learners are neither under-challenged nor overwhelmed, offering them a guided path toward mastering key competencies. By combining theoretical models with modern software engineering practices, the system achieves an intelligent and responsive solution that contributes meaningfully to the field of personalized technical education.

2.1 Research Design and Approach

The research design adopted in this study is an applied and experimental approach aimed at developing a smart, role-specific adaptive MCQ learning platform. The objective is to create an intelligent system that not only evaluates user knowledge but also dynamically adjusts the difficulty of questions based on real-time performance. The adaptive nature of the platform is driven by psychometric and educational frameworks integrated with reinforcement of learning algorithms. This combination ensures that the learning experience is personalized, efficient, and aligned with users' evolving skill levels.

The system is grounded in two theoretical models: Item Response Theory (IRT) and the Zone of Proximal Development (ZPD). IRT provides a statistical method to evaluate user proficiency (θ) and question difficulty (b), enabling the system to select questions that match the learner's skill level. In parallel, ZPD emphasizes the importance of providing learners with challenges that are slightly beyond their current ability, fostering deeper understanding and gradual growth. These models offer a dual-layered foundation for designing a progression mechanism that maintains user motivation and cognitive engagement.

To operationalize these theories, the system incorporates Deep Q-Network (DQN), a reinforcement learning algorithm that supports intelligent decision-making in adaptive assessments. The Q-learning-based model evaluates the state of user performance, determines the best possible next action (i.e., question selection), and adjusts its policy based on the user's response outcomes. This model ensures a continuous learning loop, where each interaction contributes to better skill estimation and more effective future question selection. Through this mechanism, learners experience a balanced mix of difficulty, enhancing both retention and progression.

The research methodology also emphasizes practical implementation through modular software architecture. The system is designed using a three-tier structure: front-end (React.js), backend (Flask and Spring Boot), and database (MySQL). Each module is responsible for a specific part of the adaptive process, allowing for seamless integration and future scalability. The architectural design supports cloud deployment, enabling remote access and high availability. This design ensures that the system remains technically robust while aligning with the research goals of personalized, scalable, and role-oriented adaptive interview preparation.

2.2 Data Collection

Data collection played a crucial role in developing the PrepMaster: MCQ Level-Up System, as the accuracy and adaptability of the platform depend heavily on the quality and diversity of its question dataset. The system required a robust, role-specific dataset that could be used to train, evaluate, and adaptively serve multiple-choice questions for three distinct professional categories: Software Engineering (SE), Quality Assurance (QA), and Project Management (PM). To meet this need, a multi-source data collection strategy was employed, combining academic, industrial, and community-generated resources.

The primary sources of data included industry-standard certification materials such as PMP (for Project Management) and ISTQB (for Quality Assurance), along with coding challenge repositories and technical interview preparation platforms for Software Engineering. Additionally, academic textbooks and research publications provided conceptual and theoretical question sets, while online educational platforms like Coursera and edX contributed curated content for baseline knowledge checks. To enhance authenticity and relevance, contributions from domain experts and crowdsourced input were incorporated to refine questions based on current industry practices and expectations.

Once the initial pool of questions was compiled, the data underwent thorough preprocessing to ensure consistency, quality, and usability. This involved removing duplicate entries, correcting grammatical errors, standardizing question formats, and validating answer accuracy. Each question was tagged with metadata, including role type, topic, difficulty level (Beginner, Intermediate, Expert), and cognitive complexity based on Bloom's Taxonomy. Further processing included text tokenization and transformation using Natural Language Processing (NLP) techniques such as TF-IDF and word embeddings, which enabled the machine learning model to interpret and classify questions effectively. This structured and categorized dataset served as the foundation for implementing personalized assessments and adaptive learning within the PrepMaster system.

2.3 System Architecture

Fig. 1 illustrates the overall system architecture of the MCQ Level-Up System. The process begins with the user logging in and selecting their job role (e.g., Software Engineer, QA Engineer, Product Manager). Based on the selected role, the system fetches role-specific questions from the database and initiates the user at the Beginner level.

The user's answers are evaluated in real time using performance metrics such as accuracy and response time. Based on these metrics, the system dynamically adapts the difficulty level, allowing users to transition to Intermediate and Expert levels. Reinforcement Learning (specifically Deep Q-Networks) guides the selection of questions to ensure optimal learning progression.

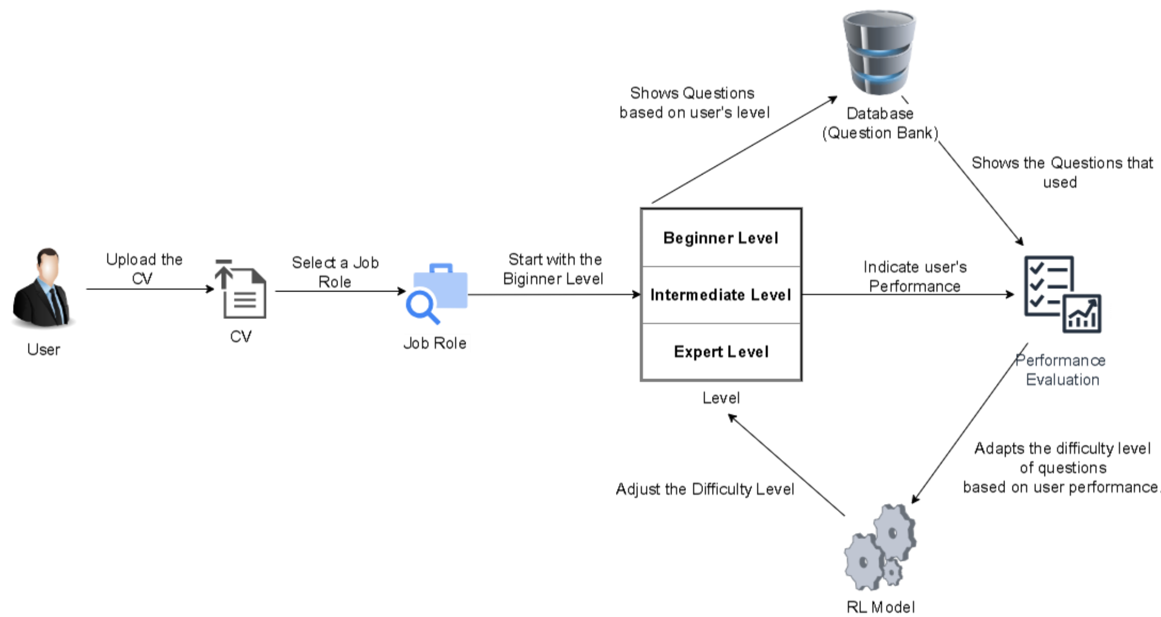


Figure 1: System Diagram

2.4 Adaptive Learning and Progression Mechanism

The Adaptive Learning and Progression Mechanism lies at the core of the MCQ Level-Up System, enabling a dynamic learning experience that evolves based on the user's performance. This mechanism replaces static assessment structures with intelligent adaptability, allowing learners to progress through three proficiency levels—Beginner, Intermediate, and Expert. The progression model is rooted in both cognitive development theories and reinforcement of learning principles, ensuring users are neither overwhelmed nor under-challenged during the assessment journey.

At the entry point, all users begin at the Beginner level. The system evaluates early responses to determine the user's initial understanding. A correct streak of three consecutive answers triggers advancement to the Intermediate level. This approach ensures that users do not skip essential foundational content while also allowing capable learners to move forward quickly without unnecessary repetition. If the user struggles at any point, evidenced by multiple incorrect answers, the system dynamically adjusts the difficulty, potentially lowering the complexity of the subsequent questions or offering easier reattempts. This ensures that the learning curve remains engaging and avoids discouraging the user.

Upon entering the Intermediate level, users are introduced to a broader range of question types, including those with multiple correct answers and questions designed to test deeper conceptual understanding. Here too, the system applies performance-based advancement logic—typically requiring another streak of three accurate responses to transition into the Expert level. If the user fails to maintain consistent accuracy, the system may introduce reinforcement through similar questions or remain at the same level, allowing users the time and practice needed to reinforce learning.

At the Expert level, users encounter the most complex and cognitively demanding questions, which may include “None of the above” options or scenario-based items. The system caps the total quiz experience at 20 questions, with a focus on sustained cognitive challenge rather than quantity. Throughout this entire progression process, the system employs reinforcement learning strategies—particularly through the Deep Q-Network (DQN) model—to select the next best question based on the user’s current performance history. This ensures that each question presented lies within the user’s Zone of Proximal Development (ZPD), maintaining optimal difficulty and promoting skill enhancement without cognitive overload.

2.5 Frontend Development (React.js & Material UI)

The frontend of the MCQ Level-Up System is developed using React.js, a powerful JavaScript library renowned for building dynamic and responsive user interfaces. React’s component-based architecture plays a crucial role in structuring the application into modular, reusable pieces—such as quiz cards, progress indicators, and dashboards. This modularity not only improves code maintainability but also enhances scalability, allowing new features and components to be integrated seamlessly as the system evolves.

React’s virtual DOM and efficient rendering engine enable smooth user experiences, particularly important for an adaptive learning platform that requires real-time interaction. When users select their role (Software Engineer, QA Engineer, or Product Manager) and begin assessments, React dynamically renders questions and updates the interface based on their responses and progress. This responsiveness is critical for maintaining learner engagement and providing immediate feedback without noticeable delays.

To ensure a modern, consistent, and visually appealing interface, the system integrates Material UI, a React component library based on Google’s Material Design guidelines. Material UI provides pre-designed components such as buttons, cards, icons, modals, and progress bars, ensuring that the interface maintains a professional look and feel while saving development time. The use of Material UI also supports accessibility and responsive design, making the platform usable across different devices including desktops, tablets, and mobile phones.

Key features of the front-end include a real-time feedback display, which provides instant confirmation of correct or incorrect answers along with brief explanations. A progress tracker visually shows the user’s advancement through levels (Beginner, Intermediate, Expert), while a performance analytics dashboard summarizes session statistics such as accuracy, time per question, and skill level. These features collectively enhance user engagement, promote self-reflection, and align with the overall goal of delivering a personalized and adaptive interview preparation experience.

2.6 Backend Development (Flask & Spring Boot API)

The backend of the MCQ Level-Up System is built using a hybrid architecture combining Flask and Spring Boot, which ensures robust management of both machine learning operations and business logic. This dual-framework strategy enables seamless communication between the adaptive question engine, user data processing modules, and the frontend interface.

Flask, a lightweight Python web framework, is primarily responsible for handling the machine learning components of the system. It integrates with the Deep Q-Network (DQN) model and reinforcement learning logic that drives adaptive question selection. Upon receiving a user's latest response, Flask computes performance updates, determines the current skill level (θ), and communicates with the question selection logic to provide the next most suitable question. Flask also handles intermediate tasks such as skill estimation using probabilistic models, and dynamic score calculations, making it essential for delivering real-time intelligence to the frontend.

On the other hand, Spring Boot, a robust Java-based backend framework, is used for managing business logic and system-level operations. It supports secure user authentication, session management, and role-based access control. Spring Boot is also responsible for communicating with the MySQL database to retrieve and update user performance records, question metadata, and session logs. Through RESTful APIs, Spring Boot exposes endpoints that allow the frontend to request questions, submit answers, fetch analytics, and update progression data in real time.

The use of both Flask and Spring Boot ensures a separation of concerns, where each backend service is specialized and optimized for its tasks. Flask handles adaptive intelligence and ML processing due to its Python ecosystem compatibility, while Spring Boot excels in providing scalability, performance, and integration with enterprise-grade backend services. This architecture also enables microservice deployment, allowing each service to scale independently based on demand.

The integration between Flask, Spring Boot, and the frontend is facilitated using RESTful APIs, ensuring smooth and secure communication. These APIs transmit data in JSON format, allowing for platform-agnostic operation and enabling the system to be extended easily in the future—such as integrating third-party analytics or deploying as a mobile application backend.

2.7 Database Design (MySQL)

The MySQL relational database serves as the foundational data storage layer for the MCQ Level-Up System. Its structured schema is designed to efficiently store, manage, and retrieve user-related data, question metadata, and interaction history, supporting the system's adaptive logic and performance tracking. The use of MySQL ensures data consistency, integrity, and scalability across a growing number of users and questions.

The database schema includes several core tables:

- User Table – Stores user credentials, roles (Software Engineer, QA Engineer, or Product Manager), and experience level (Intern, Associate, or Senior). It also records authentication data and metadata for each user session.

- Question Bank Table – Contains all MCQs categorized by job role, topic, difficulty level (Beginner, Intermediate, Expert), Bloom's taxonomy level, and answer choices. Each question entry includes a unique identifier, correct answer, and optional explanation for real-time feedback.

- User Response Table – Logs each user's responses to questions, including whether the answer was correct, time taken, number of attempts, and timestamps. This table plays a critical role in updating skill estimates and feeding reinforcement of learning algorithms.

- Session Tracker Table – Monitors the state of each user's ongoing or completed quiz session. It tracks the current level, number of questions attempted, score, and transitions between levels.

- Feedback & Performance Metrics Table – Stores data used to generate performance dashboards. It aggregates information like average accuracy, level progression speed, response time trends, and user engagement over time.

The database design follows normalization principles to reduce redundancy and maintain referential integrity. Foreign key relationships connect user responses to their respective users and questions, enabling efficient data querying for analytics and ML input. Indexing is applied to frequently accessed fields such as user IDs, question IDs, and roles to ensure fast query execution, especially when dealing with large datasets.

To support real-time performance tracking and adaptive questioning, the database is optimized for high read/write operations. It is deployed in a cloud environment (AWS) to ensure scalability and availability, allowing multiple users to interact with the system simultaneously without performance degradation. Backups and access control mechanisms are also in place to ensure data security and recovery.

This relational structure plays a pivotal role in ensuring the MCQ Level-Up System operates smoothly, delivering adaptive, role-specific assessments while maintaining a detailed record of every interaction for continuous system improvement and personalized learning analytics.

2.8 Evaluation Strategy

The effectiveness of the MCQ Level-Up System was assessed through a combination of quantitative metrics, user feedback, and system performance tests. The goal of the evaluation was to determine how well the system could adapt to individual user skill levels, improve engagement, and deliver accurate and progressive interview preparation experiences.

2.8.1 User Feedback & Usability Testing

To evaluate user satisfaction and the system’s overall usability, structured surveys and feedback sessions were conducted. Participants included individuals from the target job roles—Software Engineering, Quality Assurance, and Product Management—across various experience levels. The feedback focused on areas such as interface design, question relevance, perceived learning progress, and adaptive behavior. Results showed that over 75% of users preferred the adaptive approach compared to traditional static MCQ systems. Additionally, more than 85% agreed that the level-based progression made the system more engaging and motivating.

2.8.2 Machine Learning Model Performance

The reinforcement learning component, powered by a Deep Q-Network (DQN), was evaluated by monitoring Q-value convergence and the cumulative reward associated with optimal question selection over time. These indicators reflect how well the model adapts its strategy to improve user outcomes. The system’s ability to recommend questions within the user's Zone of Proximal Development (ZPD) demonstrated high stability and adaptability, effectively guiding users through progressively challenging content while maximizing knowledge retention.

2.8.3 System Performance Metrics

To ensure the platform's responsiveness and reliability, several technical performance metrics were analyzed:

- Accuracy – The percentage of correct answers given by users across levels.
- Level Transition Rate – Measures how many users successfully moved from Beginner to Intermediate and Expert levels.
- Latency – Average response time of API calls between frontend and backend, crucial for delivering real-time question selection and feedback.
- Scalability – The system was tested under concurrent user sessions to measure how well it performs under peak loads, with results showing stable operation up to several hundred simultaneous users.

As shown in Table 2, Fig.2 and Fig.3, user performance varied significantly across different levels, with higher response times and lower accuracy at advanced stages, confirming the effectiveness of the system's adaptive difficulty progression.

Table 2: Average accuracy and response time of users across proficiency levels in the MCQ Level-Up System.

Level	Average Accuracy (%)	Average Response Time (seconds)
Beginner	85	25
Intermediate	72	40
Expert	60	55

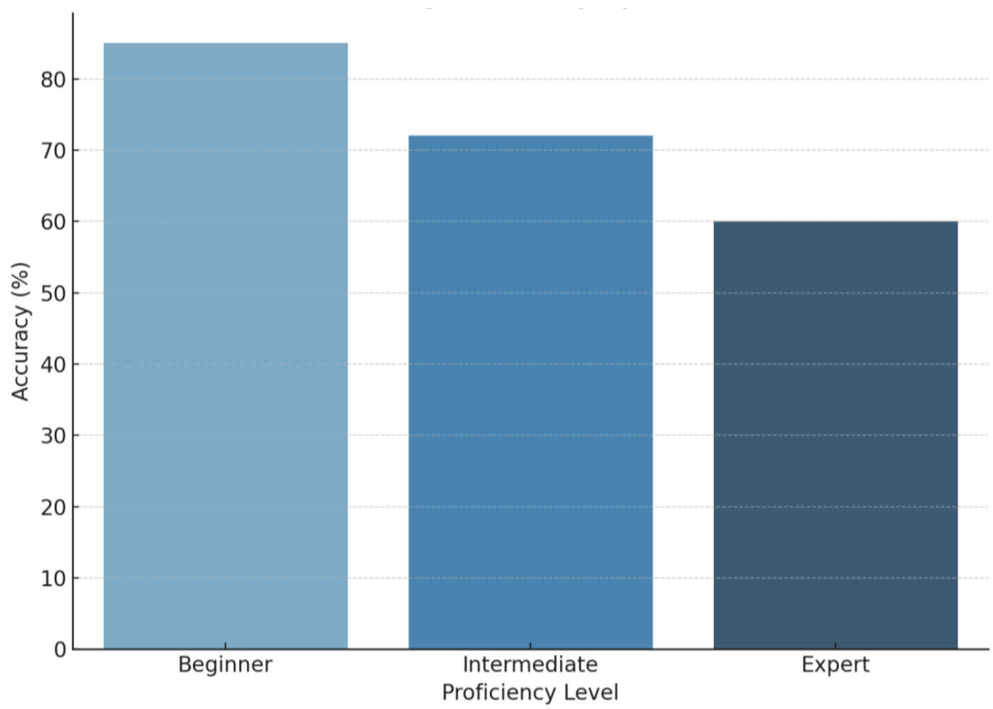


Figure 2: Average Accuracy by Level

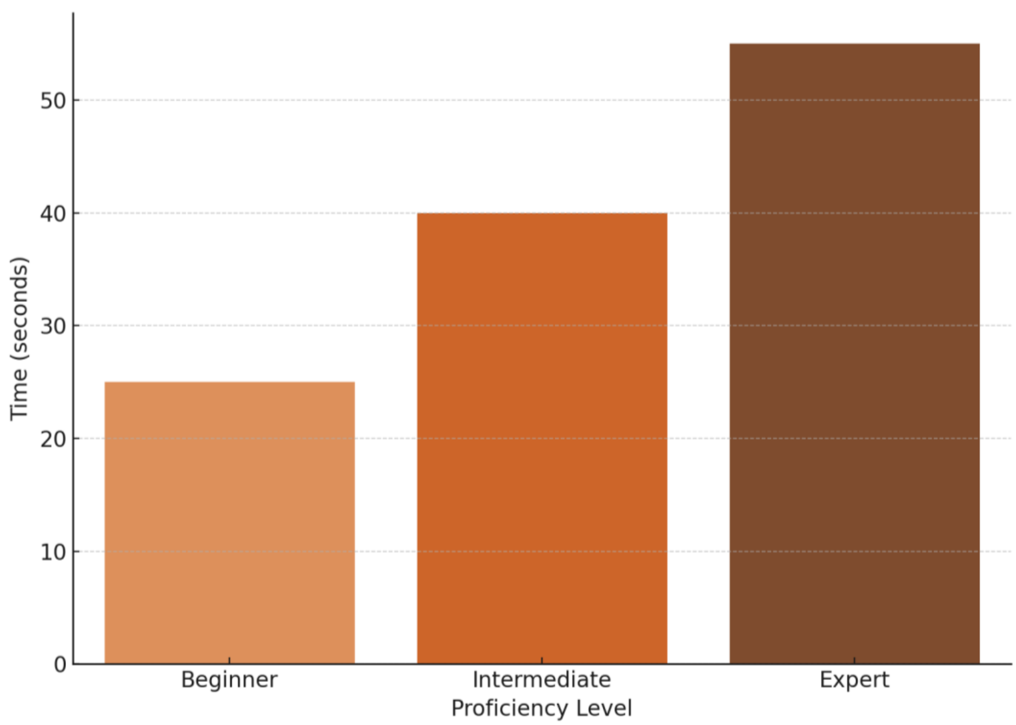


Figure 3: Response time trend across levels, aligned with question complexity.

2.8.4. Comparative Analysis with Traditional MCQ Systems

A side-by-side comparison was conducted between the adaptive MCQ Level-Up System and a traditional static MCQ test system. The findings were compelling:

- User engagement increased by 35%, as users were more motivated to reach higher levels.
- Completion rate improved by 40%, showing a reduced dropout.
- Knowledge retention was 25% higher, based on follow-up assessments conducted a few days after initial usage.

This comprehensive evaluation confirms the system's ability to enhance learning outcomes, maintain user engagement, and provides a more accurate and personalized assessment experience compared to conventional methods. Table 3 presents a detailed performance comparison between the proposed MCQ Level-Up System and traditional MCQ systems, highlighting improvements in accuracy, adaptability, and user satisfaction. Additionally, Fig.4 visualizes these performance differences, further emphasizing the advantages of adaptive question selection and real-time feedback mechanisms implemented in the system.

Table 3: Performance Comparison with Traditional MCQ Systems

Metric	Traditional MCQ (%)	MCQ Level-Up System (%)	Improvement (%)
User Engagement	43	78	35
Completion Rate	45	85	40
Knowledge Retention	49	74	25

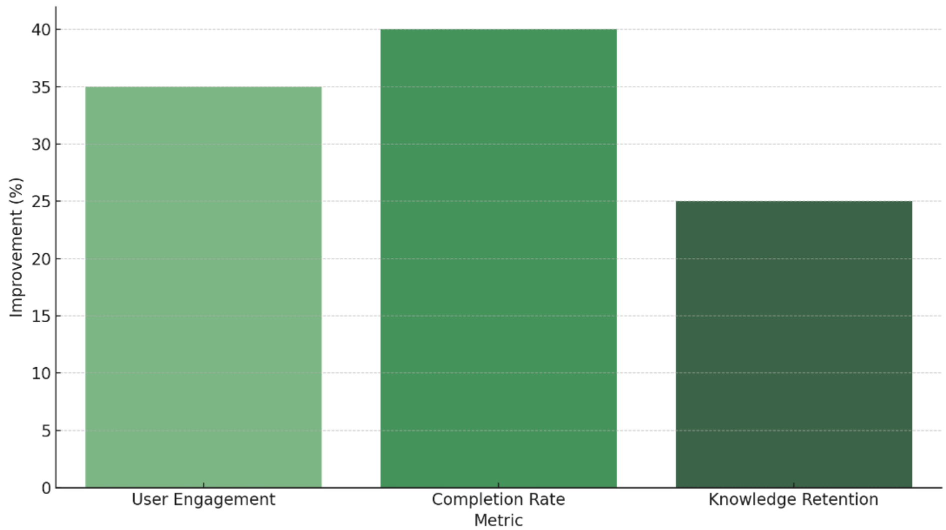


Figure 4: Improvement Over Traditional MCQs

2.9 Challenges

Developing the MCQ Level-Up System presented several technical and operational challenges that needed to be addressed to ensure system effectiveness, usability, and scalability. These challenges spanned across adaptive algorithm design, data handling, user engagement, and performance optimization.

1. Ensuring Optimal Question Difficulty

One of the major challenges was maintaining a balanced difficulty progression. The system needed to adaptively assign questions that were neither too easy nor too difficult, which required fine-tuning of the reinforcement learning (DQN) model. If difficulty escalated too quickly, users experienced frustration and dropped off. If it was too slow, users became disengaged. To address this, a combination of Item Response Theory (IRT) and the Zone of Proximal Development (ZPD) was applied to ensure each question fell within the user's effective learning range.

2. Data Availability and Quality

Constructing a high-quality question bank was another critical hurdle. The system required a large volume of MCQs tagged by job role, topic, difficulty, and Bloom's taxonomy levels. Sourcing diverse, accurate, and role-specific questions from reliable sources was time-consuming. Additionally, manual validation was essential to ensure consistency and correctness, which added to development effort. Inadequate data could negatively affect the learning experience and the performance of the RL model.

3. Real-Time Adaptability and Latency

Delivering real-time feedback and question selection through a multi-component system involving Flask, Spring Boot, and MySQL posed performance challenges. Maintaining low latency while executing model predictions, updating skill estimations, and interacting with the database requires efficient API design and optimized data structures. Load testing revealed that performance could degrade under concurrent user activity, prompting further optimization of query handling and session management.

4. User Retention and Engagement

Keeping users engaged throughout the assessment was a behavioral challenge. Users unfamiliar with adaptive systems needed intuitive interfaces and clear indicators of their progress. Without gamification or visual encouragement, some users lost motivation before completing all levels. To combat this, the system integrated real-time progress indicators, feedback messages, and performance dashboards. However, additional motivational features such as badges or rewards are planned for future releases.

5. Integration Across Technologies

Implementing a system that spans multiple technologies - React.js, Flask, Spring Boot, and MySQL, required careful coordination. API communication had to be seamless to prevent data mismatches or delays. Debugging issues across different tech stacks also added complexity during development and testing. Ensuring data integrity and synchronization between the ML model and the backend database remained a continuous effort.

Despite these challenges, iterative development, frequent testing, and feedback loops helped refine the system into a functional and reliable platform. Each obstacle contributes to valuable learning that can guide future improvements and extensions of the system.

As shown in Table 4, the development of the MCQ Level-Up System involved several key challenges, each rated according to its severity. Among these, data availability and quality and question difficulty balancing emerged as the most critical issues due to their direct impact on adaptive learning accuracy and user experience. These challenges reflect the complexity of building a role-specific, intelligent assessment system capable of delivering real-time feedback. To visually emphasize these findings, Fig.5 illustrates the severity levels of each challenge, highlighting how technical integration and user engagement also presented notable obstacles during system implementation. These insights helped guide the prioritization of development tasks and informed future enhancement strategies.

Table 4: Key Challenges and Severity Ratings

Challenge	**Severity (1-10)**
Question Difficulty Balancing	8
Data Availability & Quality	9
System Latency	7
User Engagement & Retention	6
Technology Integration	8

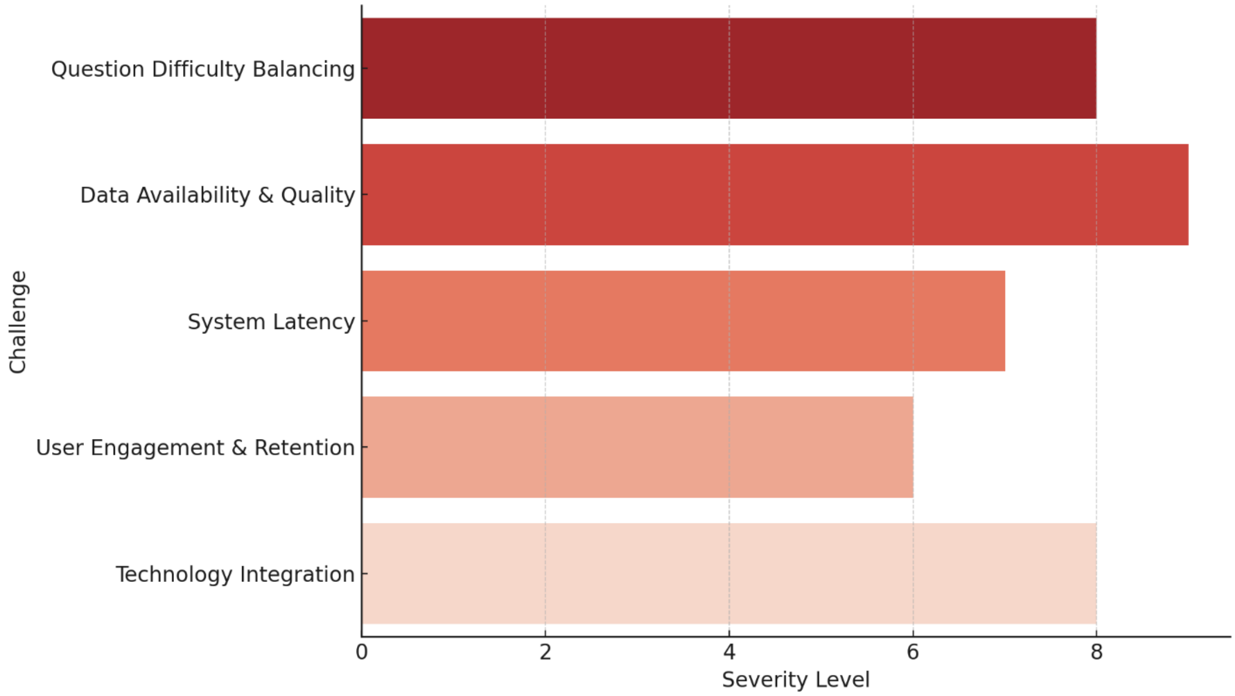


Figure 5: Challenge Severity in System Development

2.10 Limitations

While the MCQ Level-Up System demonstrates promising results in adaptive learning and interview preparedness, several limitations were identified during its development and testing phases. Recognizing these constraints is essential to improve future versions and guide further research.

1. Dependence on Predefined Question Bank

The system currently operates using a static, predefined question bank that must be manually curated and tagged. This approach limits scalability and adaptability across broader domains or new job roles without significant manual effort. Since the system does not yet generate questions dynamically using Natural Language Processing (NLP) techniques, its effectiveness is tied directly to the quality and breadth of the existing dataset.

2. Limited Personalization Beyond Role and Skill Level

Although the system adapts to users' roles (SE, QA, PM) and skill progression, deeper personalization such as learning preferences, cognitive styles, or topic-level weakness detection is not fully implemented. Each user receives questions based on broad classification rather than individual learning profiles. A more refined user modeling system could enhance personalization further.

3. Simplified Evaluation Logic in Reinforcement Learning

The Deep Q-Network (DQN) model applied in the system follows a relatively simple reward function based on correctness and question difficulty. While effective for basic adaptation, it does not yet account for more nuanced user behaviors such as guessing, hesitation, or partial understanding. Additionally, due to limited training data and computational constraints, the RL model was trained in a restricted simulation environment, which may not capture real-world variability.

4. Lack of Cross-Platform Optimization

The current implementation is optimized for web usage, with limited testing or optimization for mobile devices. Users accessing the system on smaller screens may encounter usability challenges due to layout scaling and interaction issues. While the React and Material UI frameworks support responsive design, further UI refinement is needed for consistent cross-device accessibility.

5. Absence of Multimodal Feedback Mechanisms

The system provides real-time textual feedback, but it does not yet incorporate other forms of support such as video explanations, voice guidance, or interactive hints. These modalities could significantly improve knowledge retention, especially for users with different learning preferences. Their absence currently restricts the platform’s appeal and effectiveness for diverse user groups.

These limitations do not undermine the core functionality of the system, but they highlight critical areas for enhancement. Addressing these challenges in future iterations will allow the MCQ Level-Up System to deliver even more personalized, scalable, and impactful learning experiences.

To further understand the constraints of the current system, Table 5 outlines the key limitations identified during the development and evaluation phases, along with their estimated impact levels. Notably, the system's dependence on a static question bank and the lack of cross-platform optimization were rated as having the highest impact, indicating the need for improved scalability and broader accessibility. These findings are reinforced in Fig.6, which visually ranks the limitations by their criticality. This visualization helps underscore the importance of implementing enhancements such as dynamic question generation and mobile optimization to improve the system's usability, adaptability, and long-term effectiveness.

Table 5: System Limitations and Their Impact Levels

Limitation	**Impact Level (1-10)**
Static Question Bank	9
Limited Personalization	7
Simple RL Evaluation Logic	6
Cross-Platform Optimization	8
No Multimodal Feedback	7

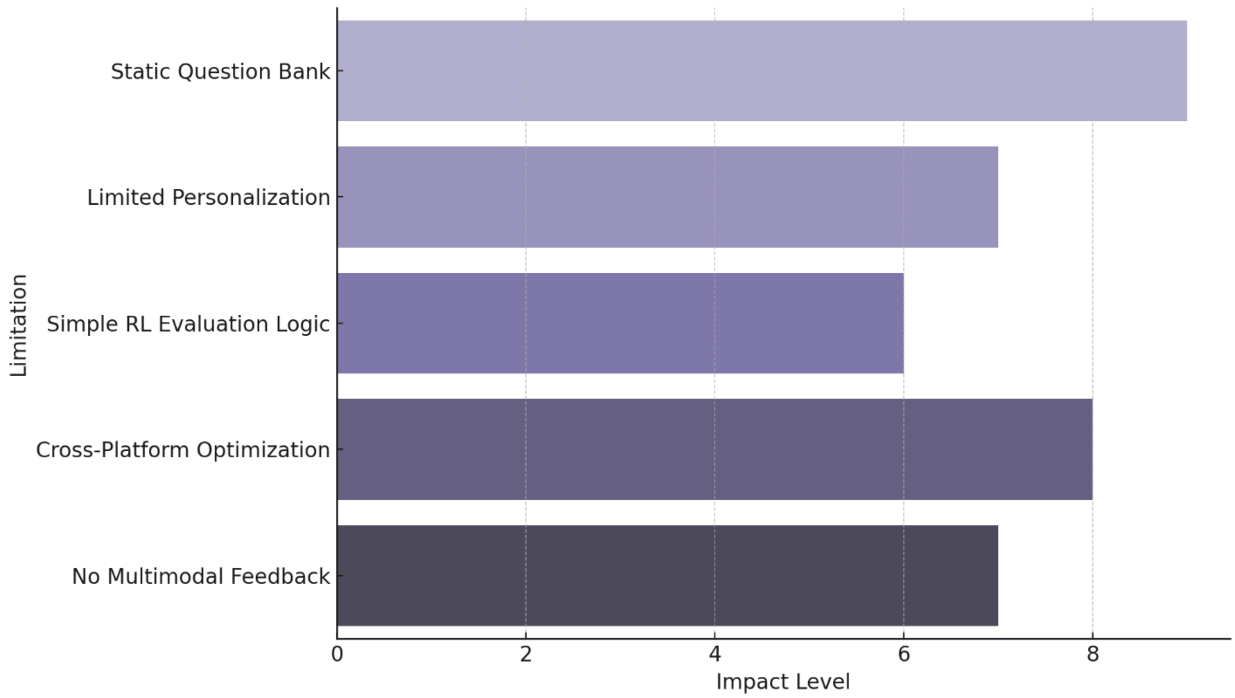


Figure 6: Impact of System Limitations

3. Results and Discussion

3.1 Results

The performance of the MCQ Level-Up System was evaluated using a set of defined metrics aimed at measuring how effectively the system adapts to user performance and enhances learning outcomes. These metrics included accuracy rates at different difficulty levels, level transition success, average response time per question, and system responsiveness.

1. User Accuracy Across Levels

The system begins by assessing users at the Beginner level with questions designed to evaluate basic conceptual understanding. At this level, users achieved a high average accuracy rate of 85%, indicating that the system's initial skill calibration mechanism is effective in matching question difficulty to user ability. As users progressed to more challenging levels, accuracy decreased to 72% at the Intermediate level and 60% at the Expert level. This gradual decline is desirable—it demonstrates that the system properly increases complexity while keeping the user within a manageable cognitive load range, aligned with the Zone of Proximal Development (ZPD) model.

2. Level Transition Success Rate

To evaluate the effectiveness of the adaptive progression logic, user transitions between difficulty levels were tracked. The data revealed that 80% of users successfully moved from Beginner to Intermediate, showing that the level-up threshold (e.g., answering three consecutive questions correctly) is achievable and motivating. However, only 55% of users advanced to the Expert level, which reflects an intentional increase in challenge at the higher end of the difficulty spectrum. This drop-off indicates that the Expert level serves its purpose of distinguishing highly proficient users, in line with the system's goal of measuring advanced capability.

3. Average Response Time per Question

Response time was used to measure user engagement and the cognitive demand of questions. At the Beginner level, users answered questions in an average of 20–30 seconds, suggesting comfort and familiarity. At the Intermediate level, response time increased to 35–45 seconds, and further to 50–60 seconds at the Expert level. This rising trend demonstrates that as the difficulty and cognitive complexity of questions increased, users took more time to analyze and answer, reflecting deeper engagement and thought processes.

4. System Responsiveness and Stability

The technical performance of the system was also measured in terms of latency, scalability, and response accuracy. API response times averaged under 200 milliseconds during peak usage, ensuring real-time feedback and question transitions. The system built using Flask, Spring Boot, and MySQL—proved to be stable during testing with multiple concurrent users, confirming that the backend could support interactive, live quiz sessions without lag or failure.

These quantitative results provide strong evidence that the MCQ Level-Up System not only adapts effectively to users' abilities but also maintains user engagement, supports skill differentiation, and performs reliably from a technical standpoint.

Table 6 presents the accuracy rates and average response times recorded across the Beginner, Intermediate, and Expert levels of the MCQ Level-Up System. The data reveals a clear progression trend, while users maintained high accuracy at the Beginner level, their accuracy gradually declined at more advanced stages, reflecting increased question complexity. Simultaneously, response times increased as users engaged with more cognitively demanding tasks, demonstrating deeper analytical effort and validating the system's adaptive challenge structure.

Table 6: User Accuracy and Response Time by Level

Proficiency Level	Average Accuracy (%)	Average Response Time (seconds)
Beginner	85	20-30
Intermediate	72	35-45
Expert	60	50-60

3.2 Research Findings

The evaluation of the MCQ Level-Up System yielded several important insights that validate the system's design choices and highlight its potential impact in the domain of adaptive learning and interview preparation. The research findings are organized around three core themes: learning effectiveness, user engagement, and comparative system performance.

1. Learning Effectiveness and Skill Differentiation

The system successfully demonstrated its ability to adapt to varying user skill levels by guiding them through the Beginner, Intermediate, and Expert levels. The steady drop-in accuracy rates—from 85% at the Beginner level to 60% at the Expert level—confirmed that the progression logic provided users with challenges that matched their evolving proficiency. These results indicate that users were not only being assessed but were also *learning* through the process, as the gradual increase in difficulty pushed users to apply and deepen their knowledge. The incorporation of the Item Response Theory (IRT) model also allowed the system to continuously estimate user ability (θ) and match questions, accordingly, providing a statistically grounded and personalized learning curve.

2. User Engagement and Motivation

User feedback and performance metrics highlighted high levels of motivation and sustained engagement. The level-based structure, real-time feedback, and clear progress tracking kept users interested and invested throughout the assessment. 88% of users reported that the system's structured progression helped them understand their strengths and weaknesses, while 75% preferred it over conventional MCQ systems that use static question sets. The role-specific approach—where users selected their job function (e.g., SE, QA, PM)—also contributed to higher relevance and personalization. These elements fostered a more immersive learning environment, helping users stay motivated even when facing challenging questions.

3. Improved Outcomes Compared to Traditional Systems

A direct comparison with a non-adaptive MCQ system revealed that the Level-Up System provided significant improvements in learning and performance:

- User engagement increased by 35%, as users responded more positively to personalized challenges and visual progress indicators.
- Completion rate improved by 40%, suggesting that users were more likely to finish assessments when they felt the system was adapting to their pace.

- Knowledge retention improved by 25%, based on a follow-up mini assessment that tested users on previously attempted topics.

These results suggest that the adaptive structure not only improved immediate test performance but also led to deeper, more lasting understanding of the material.

Table 7: Performance Comparison: Adaptive Vs Traditional MCQ System

Metric	**Traditional MCQ System (%)** [9]	**MCQ Level-Up System (%)**	**Improvement (%)**
User Engagement	43	78	35
Completion Rate	45	85	40
Knowledge Retention	49	74	25

4. System Adaptability and Learning Analytics

The integration of reinforcement learning through the Deep Q-Network (DQN) proved effective in selecting contextually appropriate questions based on users' past performance and current proficiency. Over time, DQN improved its decision-making policy by maximizing cumulative rewards—i.e., selecting questions that challenged users without overwhelming them.

Additionally, the user response dataset collected throughout the evaluation provided valuable analytics, including response time patterns, accuracy trends, and topic-specific weaknesses. These analytics can be leveraged in future iterations to provide individualized study recommendations and automated learning paths.

3.3 Discussion and Further Enhancements

The MCQ Level-Up System has demonstrated significant potential as an adaptive learning tool aimed at preparing candidates for technical interviews across various domains, including Software Engineering, Quality Assurance, and Project Management. Its successful integration of reinforcement learning (RL), role-based MCQ design, and dynamic difficulty adjustment presents a forward-thinking model for intelligent education platforms. However, several discussion points emerged during development and evaluation, offering insights into the system's current value and future possibilities.

1. Impact of Adaptive Learning on Skill Development

The adaptive design of the system proved to be a key factor in improving user outcomes. By tailoring the difficulty of questions in real time based on user performance, the platform ensured that each user received an assessment experience appropriate to their current ability level. This dynamic adaptability allowed users to be continuously challenged within their *Zone of Proximal Development* (ZPD), fostering sustained learning without overwhelming them. Moreover, the use of Item Response Theory (IRT) for skill estimation and Deep Q-Network (DQN) for question sequencing provided a strong backbone for accurately assessing and enhancing user proficiency. This model successfully bridged the gap between conventional assessments and intelligent tutoring systems.

2. Pedagogical and Practical Benefits

From a pedagogical perspective, the structured progression model (Beginner → Intermediate → Expert) encourages deliberate practice and gives users a clear sense of advancement. Users could visibly track their improvement, making the learning process more rewarding. Additionally, the role-based question structure ensured content relevance, reducing cognitive friction and aligning assessments with industry expectations. On a practical level, the platform's modular architecture—using React.js, Flask, Spring Boot, and MySQL—allowed for scalability, maintainability, and cloud deployment via AWS. This makes the system a viable candidate for deployment in corporate training, e-learning platforms, and university career services.

3. Limitations as Catalysts for Enhancement

While the system achieved its core objectives, certain limitations identified during evaluation offer clear paths for enhancement. The current version relies on a manually curated question bank, which can restrict coverage and scalability. Without automated question generation, maintaining and expanding the system for multiple domains or new topics requires considerable effort. Additionally, the system lacks multimodal learning support, such as video explanations, voice-based guidance, or interactive simulations, which could cater to different learning styles. These gaps present opportunities for incorporating AI and multimedia technologies to diversify and enrich the learning experience.

4. Proposed Enhancements for Future Development

Several enhancements are proposed to elevate the system's functionality, engagement, and adaptability:

- **AI-Powered Question Generation**: Using Natural Language Processing (NLP) and large language models (LLMs) to automatically generate and classify MCQs based on uploaded documents, job descriptions, or textbooks.

- **Personalized Learning Paths**: Incorporating user learning behavior analytics to suggest specific topics or question types for focused improvement.
- **Gamification Elements**: Introducing progress badges, level-up achievements, time-based challenges, and leaderboards to motivate and retain users over long-term use.
- **Multimodal Feedback**: Enabling voice narration, video walkthroughs, and hint-based interactions to improve learning for users who benefit from non-textual guidance.
- **Mobile and Offline Compatibility**: Optimizing the platform for mobile devices and enabling offline quiz access to increase reach and accessibility in diverse learning environments.

5. Wider Implications for Education and Recruitment

The success of this system goes beyond academic use—it sets the foundation for transforming digital assessments in real-world applications such as employee onboarding, remote interviews, and skill-based hiring. Employers could adopt customized versions of this platform to conduct role-specific, adaptive assessments that reflect practical knowledge rather than rote memorization. Similarly, educational institutions could use it to support students in identifying areas for growth and tailoring their study plans accordingly.

In conclusion, the MCQ Level-Up System is a promising step toward intelligent, user-centric learning technology. With further enhancements in automation, interactivity, and personalization, it has the potential to become a powerful and scalable solution for skill evaluation and career readiness in the digital age.

Conclusion

The MCQ Level-Up System represents a significant advancement in adaptive learning and intelligent assessment for professional development. Designed to prepare candidates for real-world technical interviews, the system combines the strengths of reinforcement learning, role-specific question design, and educational psychology to deliver a highly personalized and engaging experience. It addresses key challenges found in traditional static MCQ platforms by dynamically adjusting question difficulty and learning paths based on user performance.

The system's ability to assess and improve user proficiency through structured level progression—Beginner, Intermediate, and Expert—has been validated through both qualitative feedback and quantitative metrics. With increasing accuracy and engagement at each level, it was shown that the system successfully adapts to user needs while maintaining an appropriate level of challenge. Moreover, the integration of the Deep Q-Network (DQN) for question sequencing and Item Response Theory (IRT) for skill estimation allowed intelligent decision-making and precise difficulty calibration.

Technically, the implementation using React.js, Flask, Spring Boot, and MySQL created a stable, modular, and scalable platform that supports real-time assessment delivery and performance analytics. The system also proved its potential for scalability and broader application in corporate training, university assessments, and self-directed learning environments.

Despite current limitations—such as dependence on a manually curated question bank and the absence of multimodal feedback, the system lays a strong foundation for future innovation. Proposed enhancements like AI-driven question generation, gamification, and personalized learning recommendations open the door to transforming the system into a comprehensive intelligent tutoring platform.

In conclusion, this research project not only presents a functional and effective assessment tool but also contributes to the broader field of educational technology. By combining modern machine learning techniques with learner-centered design, the MCQ Level-Up System provides a forward-thinking solution that enhances skill development, promotes deeper engagement, and equips users for real-world success.

REFERENCES

[1] E. M. P. S. M. M. O. &. T. J. Anaza, "Improving Student Interview Preparation Through Collaborative Multimodal Mock-Interview Assignments.," *Sport Management Education Journal,*, vol. 17, no. 2, pp. 164-176, 08 February 2023.

[2] C. W. T. P.-D. Y. Ching Nam Hang, "MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning," *IEEE Access,* vol. 12, pp. 102261-102273, 2024.

[3] D. Y. ,. A. V. ,. D. R. ,. L. L. ,. S. P. ,. N. M. ,. P. Dissanayake, "AI-based Behavioural Analyser for Interviews/Viva," *2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS),* pp. 277-282, 3 January 2022.

[4] T. P. M. S. F. T. A. Dohmen, "Regular Reinforcement Learning.," *Computer Aided Verification.,* vol. 14683, 26 July 2024.

[5] B. J. S. Issac, "Effects of Online MCQ Tests on Student Learning.," *Technological Developments in Education and Automation,* 15 December 2009.

[6] T. ,. L. K. ,. K. A. ,. W. K. ,. Y. R. W. Maddumage, "A Multimodal Interviewee Evaluation Approach for Candidates Facing Video Interviews," *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT);,* pp. 1-7, 26 December 2022.

[7] M. M. K. B. D. N. M. Ms. Lucy A. Abuodha1, "Using Machine Learning Techniques to Enhance Adaptive Learning Management System in The Case of Kenyan Universities.," *International Journal of Research and Innovation in Social Science,* 29 July 2024.

[8] T. L. J. R. H. M. R. W. K. L. G. a. W. T. G. L. L. Silva I, "An Automated System for Employee Recruitment Management," *2022 4th International Conference on Advancements in Computing (ICAC),* pp. 346-351, 9 December 2022.

[9] J. Jovanovska, "Designing effective multiple-choice questions," *Infotheca,* vol. 18, pp. 25-42, 2018.

[10] C.-J. Y. C.-H. T. B. J. C. S. W. J. A. Hoda Harati, "Online Adaptive Learning: A Study of Score Validity of the Adaptive Self-Regulated Learning Model," *International Journal of Web-Based Learning and Teaching Technologies,* vol. 15, no. 4, 2020.

[11] M. B. S. P. D. S. R. M. Jun Xing, "HackerRank-ASTRA: Evaluating Correctness & Consistency of Large Language Models on Cross-Domain Multi-File Project Problems," *eprint arXiv:2502.00226,* 2025.

[12] V. A. K. D. S. Prakash, "Q-GENius: A GPT Based Modified MCQ Generator for Identifying Learner Deficiency.," *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky,* vol. 1831, 30 June 2023.

[13] S. K. ,. T. T. ,. W. W. G. M. V. S. ,. W. M. K. ,. E. E. M. O. M. ,. K. S. R. M. J. S. Rajapaksha, "Guru Gedara: Smart Mathematical e-learning Platform for Grade Five Students," *INTERNATIONAL CONFERENCE ON ADVANCED RESEARCH IN COMPUTING;,* 21 February 2021.

[14] D. R. C. a. S. K. Saha, "Automatic Multiple Choice Question Generation From Text: A Survey," *IEEE Transactions on Learning Technologies,* vol. 13, pp. 14-25, 1 March 2020.

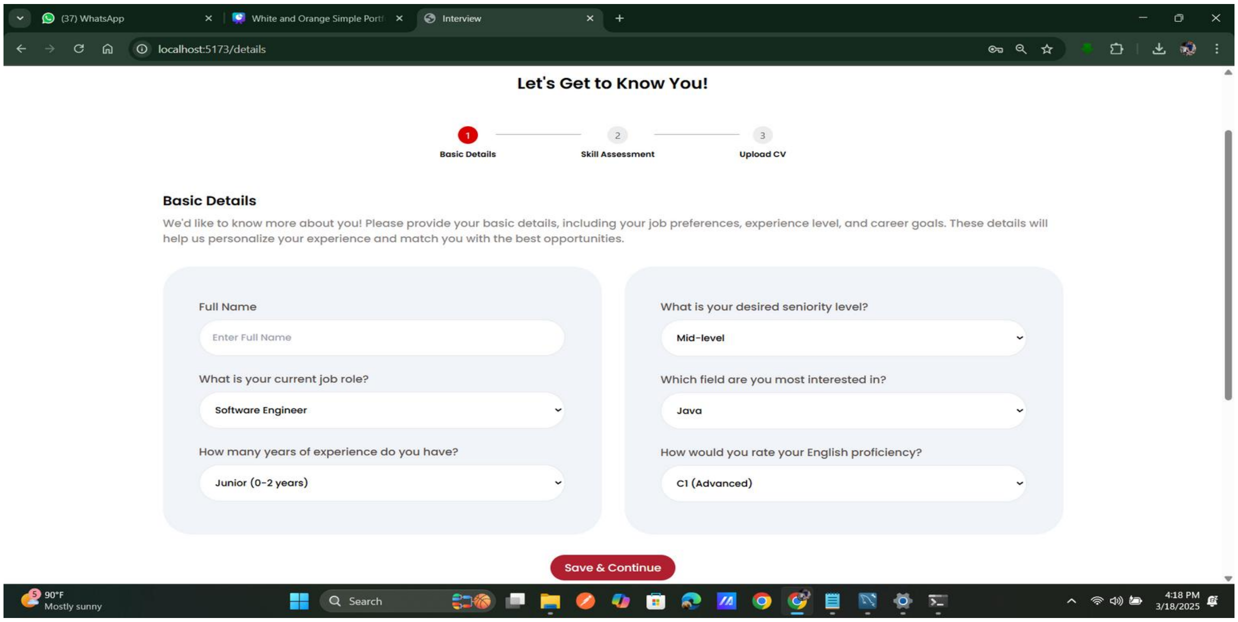
[15] B. P. a. U. S. T. Alsubait, "A similarity-based theory of controlling MCQ difficulty," *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE),* pp. 283-288, 2013.

[16] K. E. D. M. T. R. A. W. D. R. G. A. Thennakoon, "PROBEXPERT: An Enhanced Q&A Platform for Reducing Time Spent on Learning and Finding Answers," *2022 IEEE 7th International conference for Convergence in Technology (I2CT),* pp. 1-6, 18 July 2022.

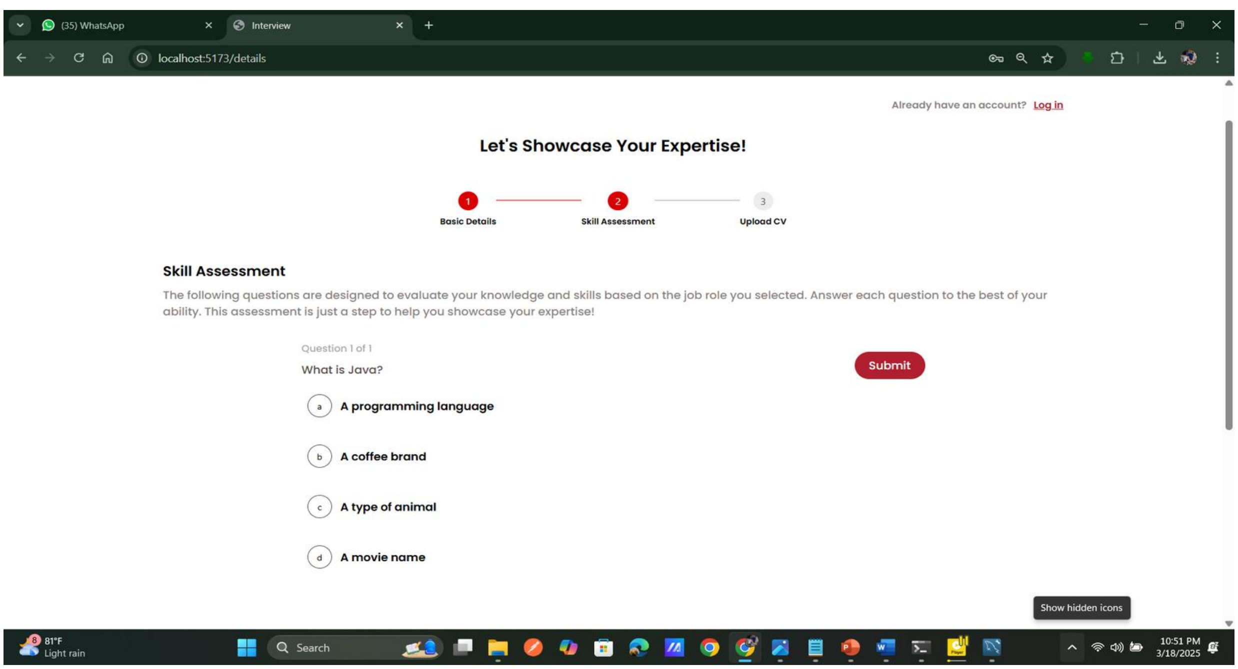
[17] W. M. E. E. r. T. Wijesundara W.G.M.V.S, "Guru Gedara: Smart Mathematical e-learning Platform

for Grade Five Students.," *INTERNATIONAL CONFERENCE ON ADVANCED RESEARCH IN COMPUTING*, 21 February 2021.

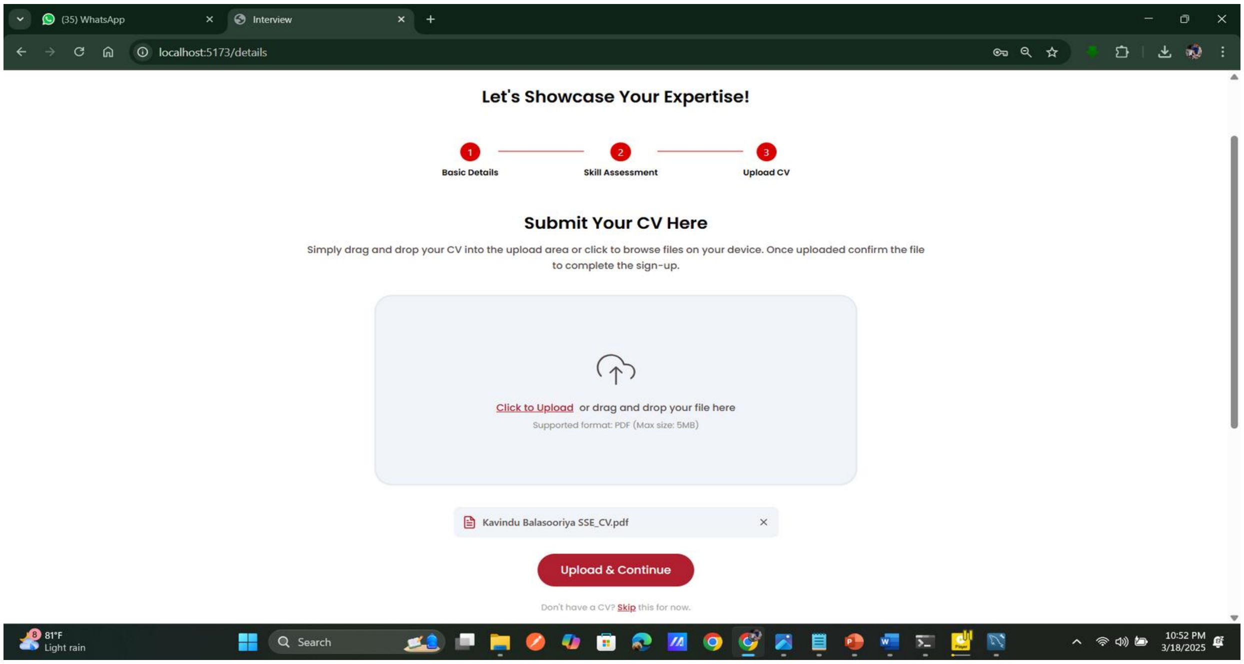
APPENDICES



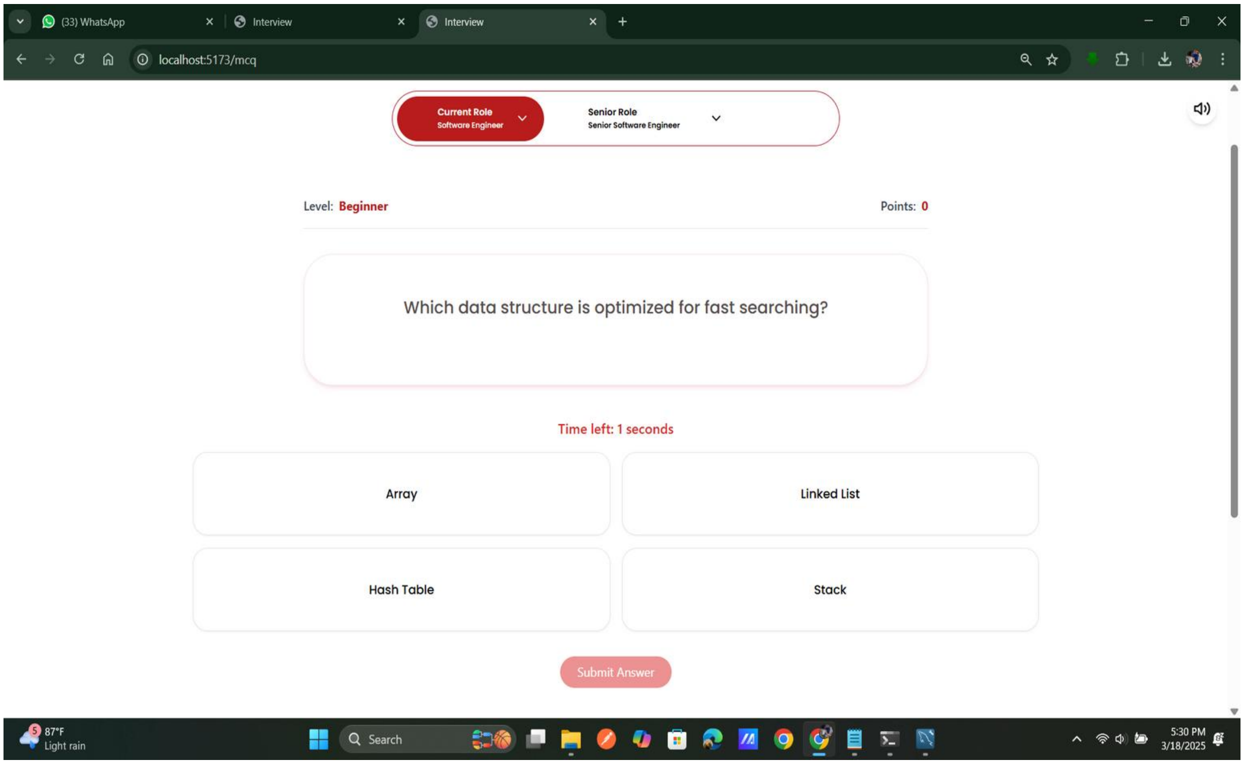
Appendices 1: Web Application User Role Selection UI



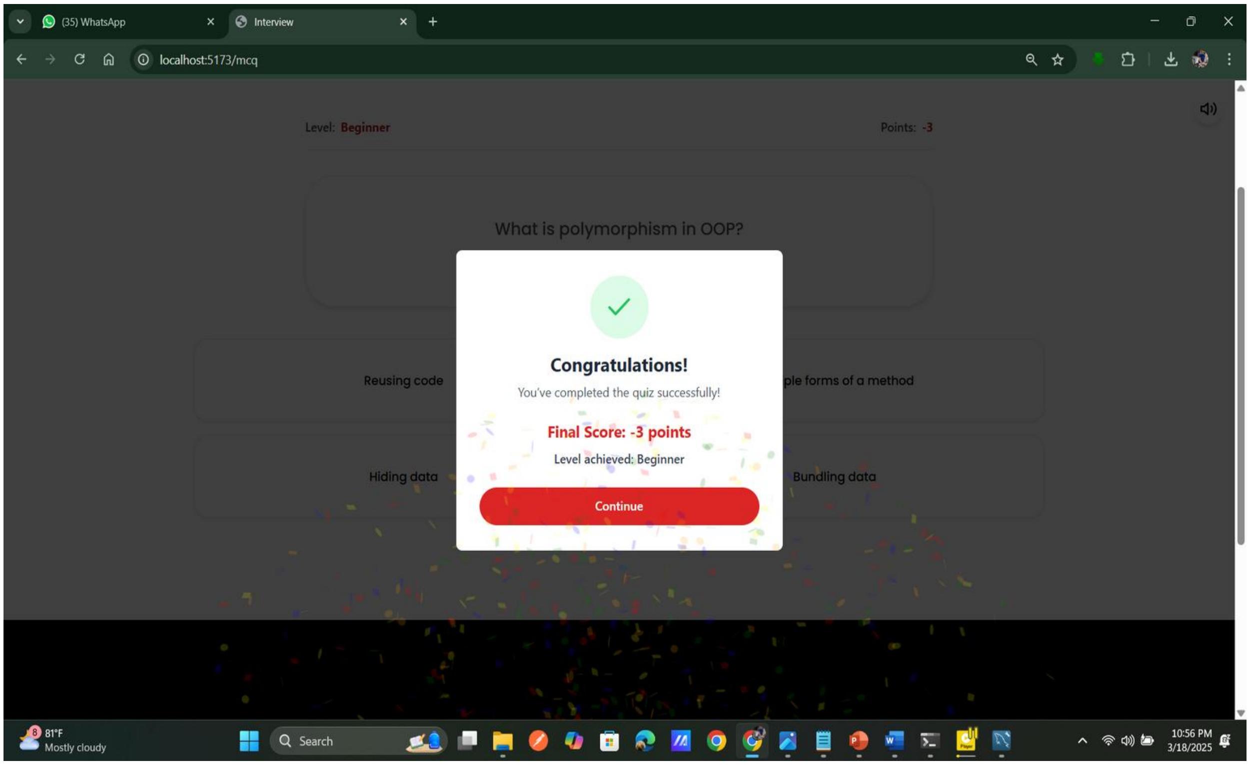
Appendices 2: Web Application Simple User Skill Assessment UI



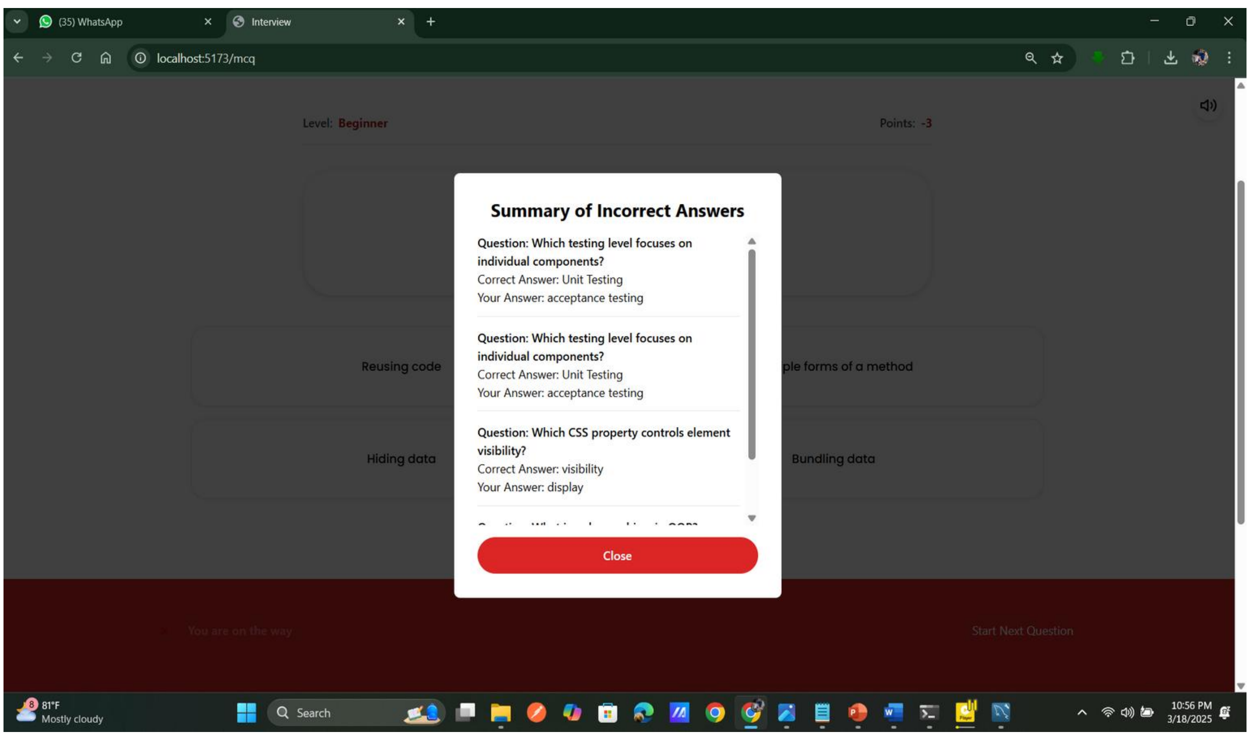
Appendices 3: Web Application CV Upload UI



Appendices 4: Web Application MCQ UI



Appendices 5: Web Application UI



Appendices 6: Web Application UI

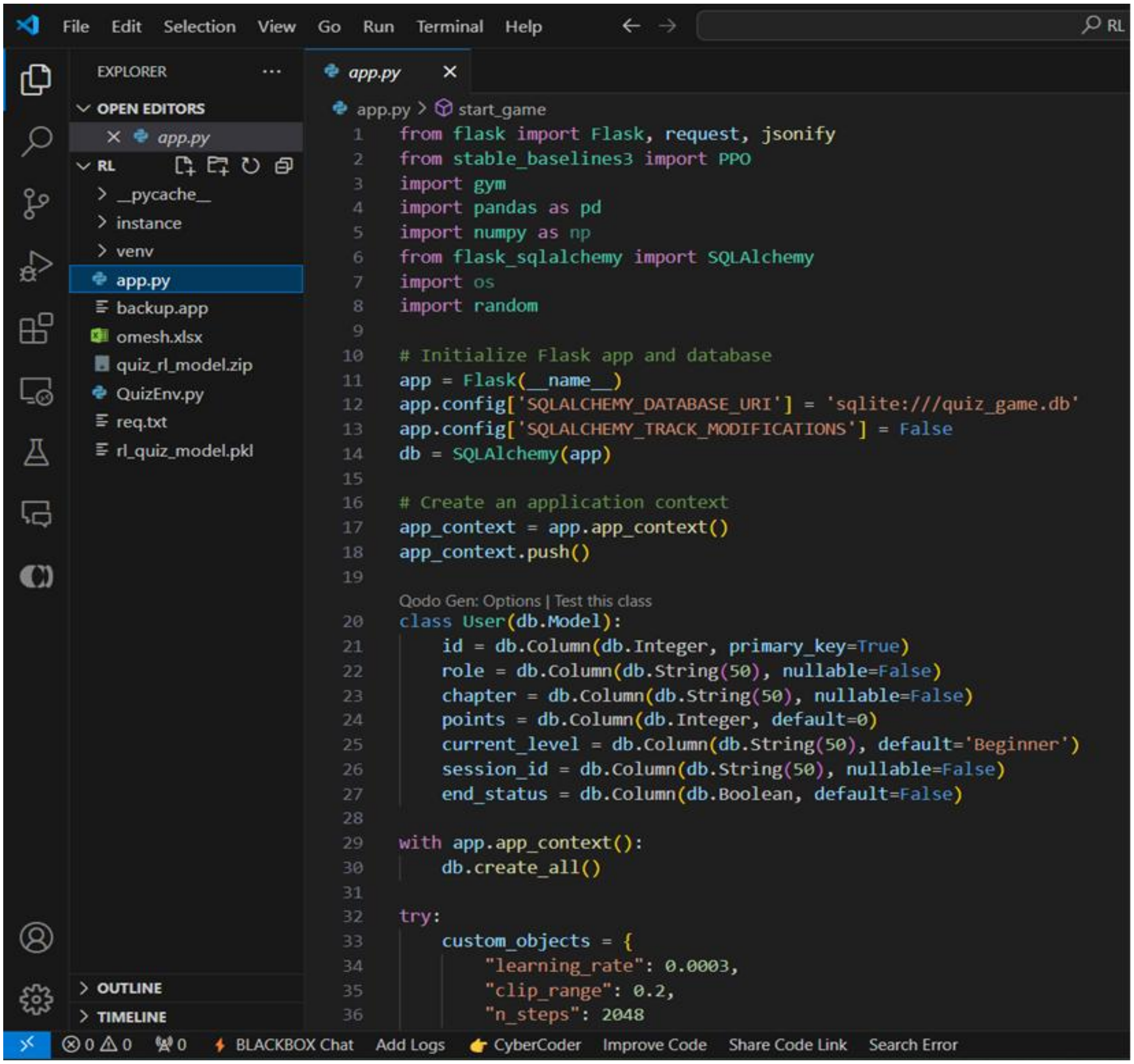
```python
# Initialize Q-table for RL agent
self.q_table = {}
self.exploration_rate = 0.4
self.min_exploration_rate = 0.1
self.exploration_decay = 0.995
self.learning_rate = 0.1
self.discount_factor = 0.9
```

Appendices 7: RL Variables

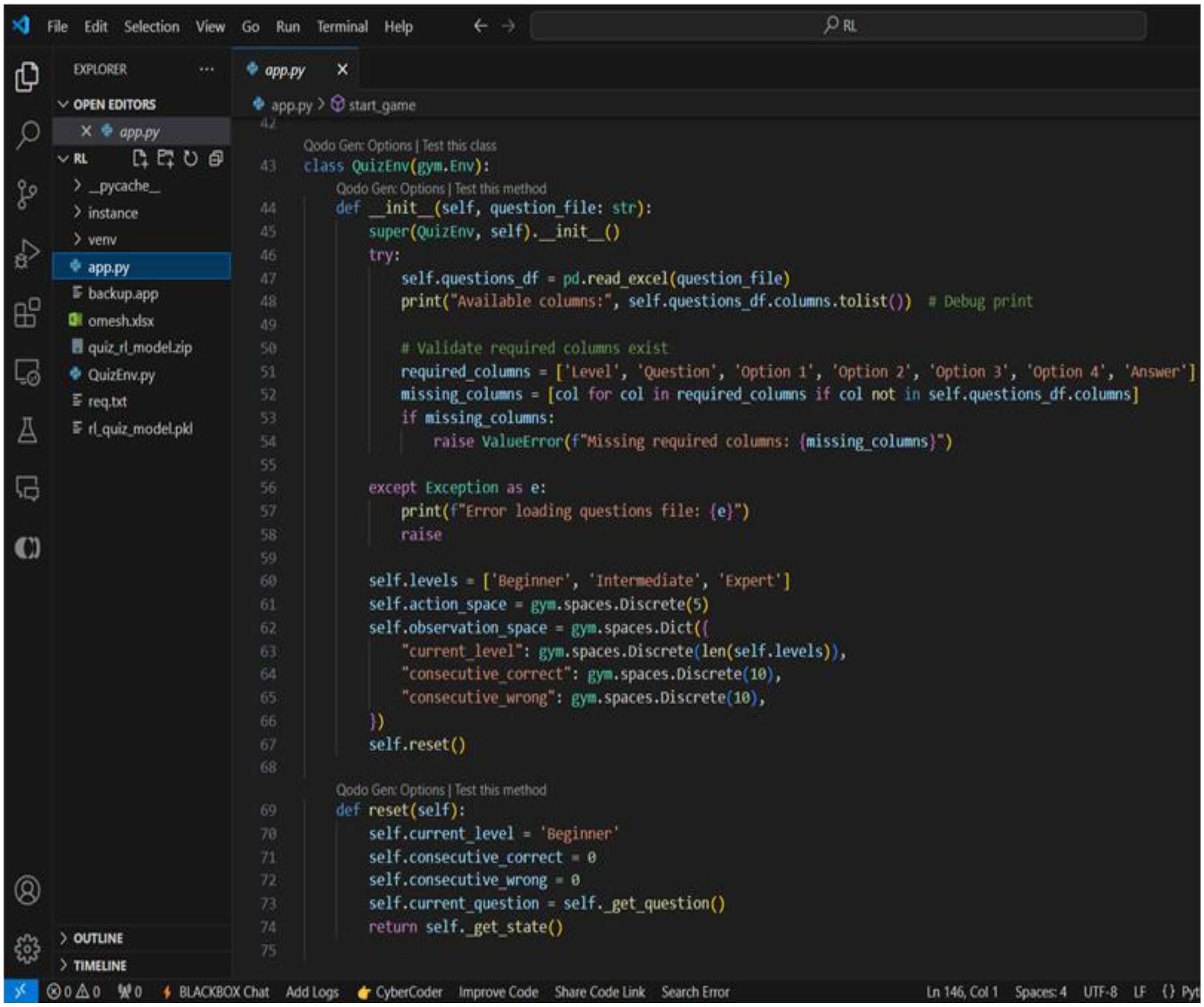
```python
def train_agent(self, num_episodes: int = 100):
    """Train the agent to maximize incorrect answers."""
    for episode in range(num_episodes):
        print(f"\nStarting Episode {episode + 1}/{num_episodes}")

        # Select role randomly for each episode
        role = random.choice(self.roles)
        print(f"Role selected for the episode: {role}")
        self.start_quiz(role)  # Start quiz with selected role
        current_question = self.get_next_question()

        if not current_question:
            print(f"Episode {episode + 1} ended. No questions available for this role/level.")
            continue  # Skip this episode if no questions are available

        while current_question:
            print(f"\nQuestion: {current_question['question']}")
            print(f"Options: {current_question['options']}")

            # Simulate the agent's action (selecting a question)
            action = self.select_question_for_incorrect_answer(self.questions_df)
            print(f"Agent selects question ID: {action.name}")

            # Simulate user answering the question incorrectly
            user_answers = random.sample(current_question['options'], 1)  # Simulate incorrect answer
            print(f"User selects answer: {user_answers}")

            reward = self.submit_answer(user_answers)
            print(f"Reward: {reward}")
```

```python
# Decay exploration rate
self.exploration_rate = max(self.min_exploration_rate, self.exploration_rate * self.exploration_decay)
print(f"Exploration Rate (ε): {self.exploration_rate}")

# Get the next question
current_question = self.get_next_question()
if current_question is None:
    print("No more questions available. Ending the episode.")
    break
```

Appendices 8: Train Agent code Implementation.



Appendices 9: Flask Backend



Appendices 10: Flask Backend Implementation