IQB PROGRAMMING QUESTION BANK WEBSITE

He ZhiPeng, Kauthar Mohd Daud

Faculty of Information Science & Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor

Abstract

Laman Web Bank Soalan Pengaturcaraan IQB ialah platform berasaskan web yang dibangunkan untuk membantu pelajar universiti, khususnya mereka yang mengambil jurusan sains komputer, dalam membuat persediaan untuk temuduga pengaturcaraan teknikal. Sistem ini menawarkan koleksi terpusat soalan pengaturcaraan yang disusun mengikut topik, tahap kesukaran dan bahasa pengaturcaraan, membolehkan pengguna melakukan carian pintar berdasarkan keperluan pembelajaran khusus mereka. Selain itu, sistem ini menyediakan ciri, menggambarkan kemajuan kajian menggunakan kalendar kajian terbina dalam, dan membolehkan interaksi rakan ke rakan melalui bahagian ulasan dan perbincangan. Fungsi ini bertujuan untuk mengatasi batasan platform sedia ada, yang sering kekurangan navigasi berstruktur, alat penjejakan yang diperibadikan dan ciri komunikasi bersepadu. Proses pembangunan telah dijalankan dalam empat fasa utama: analisis keperluan, antara muka pengguna dan reka bentuk pangkalan data, pelaksanaan sistem menggunakan teknologi seperti HTML, Vue.js, Java (SpringBoot), dan MySQL, diikuti dengan ujian. Ujian kotak hitam dan penerimaan pengguna telah dijalankan untuk memastikan kefungsian dan kebolehgunaan sistem. Keputusan daripada fasa ujian menunjukkan bahawa sistem ini mesra pengguna, berfungsi dan berkesan memenuhi keperluan pembelajaran dan amalan pengguna yang dimaksudkan.

Kata kunci: Temuduga Pengaturcaraan, Kalendar Pengajian, Buat Perbincangan.

Abstract

The IQB Programming Question Bank Website is a web-based platform developed to assist university students, particularly those majoring in computer science, in preparing for technical programming interviews. The system offers a centralized collection of programming questions that are organized by topic, difficulty level, and programming language, allowing users to perform smart searches based on their specific learning needs. In addition, the system provides features such as bookmarking questions for later review, visualizing study progress using a built-in study calendar, and enabling peer-to-peer interaction through a comment and discussion section. These functionalities aim to overcome the limitations of existing platforms, which often lack structured navigation, personalized tracking tools, and integrated communication features. The development process was carried out in four key phases: requirements analysis, user interface and database design, system implementation using technologies such as HTML, Vue.js, Java (SpringBoot), and MySQL, followed by testing. Black-box and user acceptance testing were conducted to ensure the system's functionality and usability. The results from the testing phase indicate that the system is user-friendly, functional, and effectively meets the learning and practice needs of its intended users.

Keywords: Programming Interview, Study Calendar, Make Discussion.

1.0 INTRODUCTION

In the digital age, the demand for highly skilled programmers has grown rapidly, leading to increasingly stringent standards for technical interviews, particularly in the field of computer science. One of the core challenges facing today's college students is the lack of structured and targeted resources for preparing for programming interviews. While online platforms such as LeetCode, GeeksforGeeks, and W3Schools offer practice problems and tutorials, they often have limitations such as complex navigation, subscription requirements, lack of progress tracking, and the absence of personalized feedback mechanisms. These challenges reduce the effectiveness of the learning process and create barriers for students seeking a focused and systematic approach to improving their programming skills.

To address these issues, the IQB Programming Question Bank website emerged as a web-based application, integrating interview-oriented programming questions into a user-friendly platform. The system is designed to support students majoring in computer science or related fields, as well as working programmers, in enhancing their problem-solving abilities and familiarity with common programming patterns. The platform features intelligent search and filtering functions, allowing users to categorize questions by programming language, difficulty level, and specific subject area. Users can track their progress through a learning calendar and engage in collaborative learning through the comment section.

The development of the IQB system aims to create a more structured and interactive self-learning environment tailored to the demands of real-world job interviews. By integrating common interview question types, it offers a practical and reliable alternative. The project was developed using modern web technologies, including HTML and Vue.js for the front-end interface, Java with SpringBoot and MyBatis for the back-end services, and MySQL for database management. The entire development process followed the standard software engineering lifecycle, including requirements analysis, design, implementation, and testing. The system underwent black-box testing and user acceptance testing (UAT), with participants being real students from the Faculty of Information Science and Technology (FTSM) at the National University of Malaysia (UKM).

Testing results indicated high user satisfaction, particularly in terms of interface clarity, ease of accessing questions, and the effectiveness of guiding structured programming practices. The IQB Programming Question Bank website demonstrates how carefully designed academic support tools can enhance self-directed learning processes and better prepare students for real-world technical assessments.

2.0 LITERATURE REVIEW

Currently, there are various programming learning platforms available online, catering to learners of different skill levels, including beginners, intermediate users, and advanced users. Some of the most commonly used platforms include LeetCode, W3Schools, and GeeksforGeeks. These platforms aim to help learners improve their programming skills through tutorials, examples, and programming challenges. However, each platform has its own limitations and focuses on different aspects of programming education.

LeetCode is a well-known platform that offers a large number of algorithmic problems commonly encountered in technical interviews. Users can practice problems categorized by topic and difficulty. While the platform is popular among job seekers, many of its valuable features, such as detailed solutions, filtering problems by company, and mock interviews, are only accessible through a paid subscription, making it less accessible to all students. On the other hand, W3Schools offers free web-based tutorials covering web development languages such as HTML, CSS, JavaScript, and SQL. While it is suitable for beginners, W3Schools lacks an interactive problem-solving environment and discussion community, which are crucial for in-depth learning. GeeksforGeeks is a platform that combines tutorials, programming questions, and community Q&A. However, its interface is often perceived as cluttered and intimidating, which may impact user experience, especially for newcomers.

Based on an analysis of these platforms, there is a clear gap in the market for an organized, interview-focused programming question bank that combines user-friendliness and comprehensive accessibility. The IQB Programming Question Bank website was developed to fill this gap. The system offers programming questions categorized by difficulty, topic, and programming language, with intelligent filtering capabilities. Additionally, it includes features such as bookmarking questions, tracking progress through a learning calendar, and interacting via a comment section. These designs aim to help students focus on exam preparation and effectively monitor their learning progress.

The design and development of the IQB system draw on the strengths and weaknesses of existing platforms. By integrating core functions into a simple and unified interface, IQB aims to enhance usability and accessibility for college students, particularly those preparing for technical interviews. According to research by (Spaho, Çiço, & Shabani, 2025), learning systems that integrate question tags, visual progress tracking, and peer support are more effective at maintaining long-term engagement and improving knowledge retention rates. Therefore, the IQB system is designed to provide computer science students with a focused, supportive, and organized self-learning experience tailored to their specific needs.

3.0 METHODOLOGY

This study includes requirements analysis, conceptual model design, application development, usability testing and results. The methodology describes the methods to overcome the identified problems and explains the research process carried out.

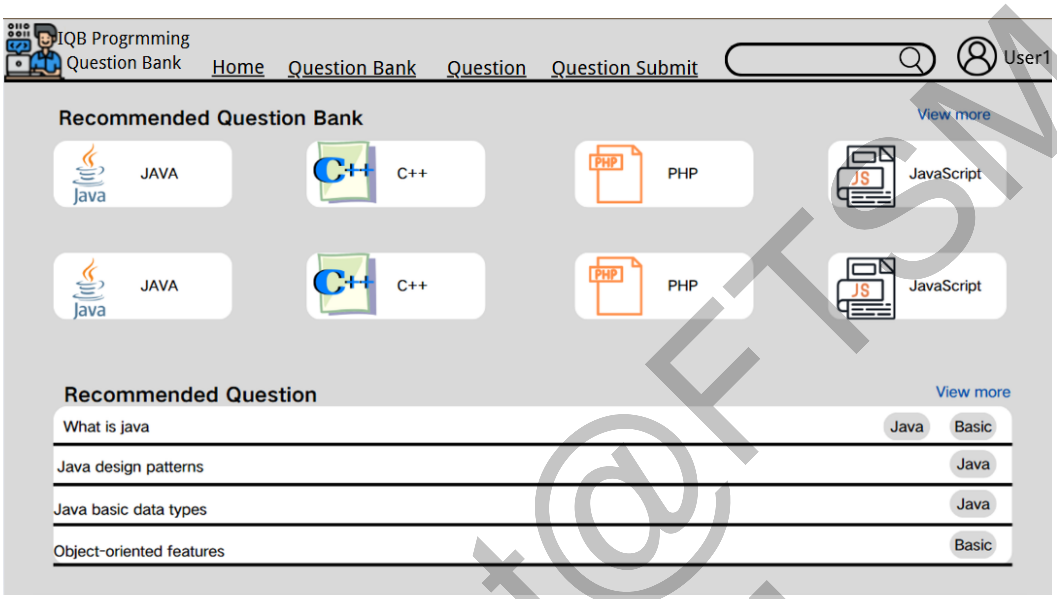
3.1 Needs Analysis

User requirements were gathered through a questionnaire distributed to undergraduate students from the Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM). The purpose of the questionnaire was to identify the main challenges students face when preparing for programming-related technical interviews and to collect suggestions for features that would be helpful in a dedicated programming question bank system.

The questionnaire included both close-ended and open-ended questions, focusing on users' current study habits, platforms they have used (e.g., LeetCode, W3Schools, GeeksforGeeks), difficulties encountered, and desired features in a new system. Based on the analysis of the questionnaire data, several key features were prioritized in system development: smart filtering by difficulty and topic, a bookmarking feature, a study calendar to track progress, and a comment section to enable peer discussion.

To assist in the process of collecting and confirming requirements, interface sketches were created to visualize the expected layout of the system and to gather early-stage feedback. These sketches include the main page layout and the filtering interface used to search questions. These visual references helped clarify how information should be organized and what interactive elements would be needed in the final design.

Figures 1 and 2 show the initial layout sketches of the system’s homepage and filtering mechanism, which were designed based on the patterns students found most helpful in other learning platforms.

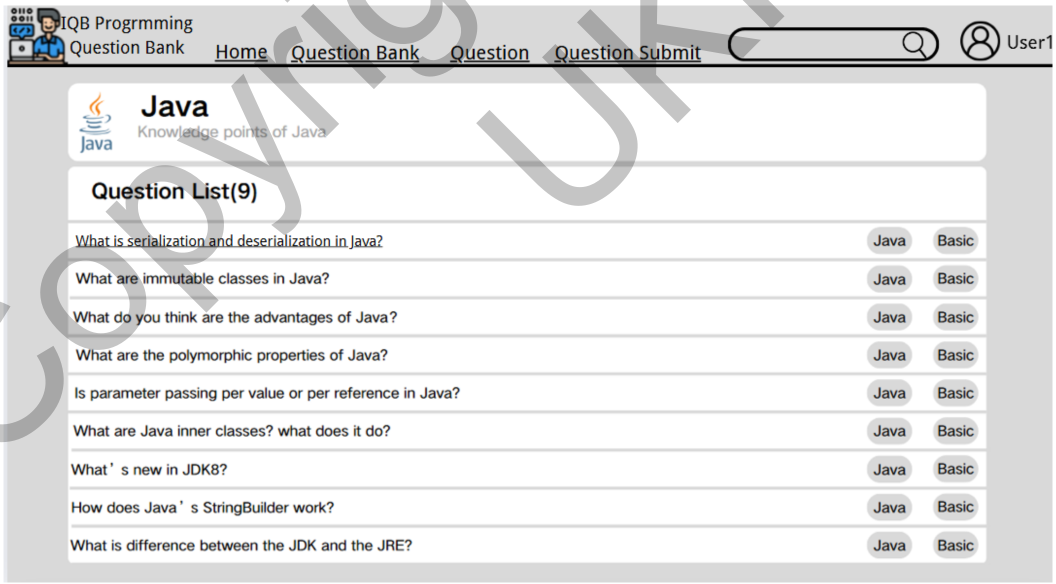


Recommended Question Bank

Recommended Question

What is java	Java	Basic
Java design patterns		Java
Java basic data types		Java
Object-oriented features		Basic

Figure 1: Homepage Interface Sketch



Java

Knowledge points of Java

Question List(9)

What is serialization and deserialization in Java?	Java	Basic
What are immutable classes in Java?	Java	Basic
What do you think are the advantages of Java?	Java	Basic
What are the polymorphic properties of Java?	Java	Basic
Is parameter passing per value or per reference in Java?	Java	Basic
What are Java inner classes? what does it do?	Java	Basic
What’s new in JDK8?	Java	Basic
How does Java’s StringBuilder work?	Java	Basic
What is difference between the JDK and the JRE?	Java	Basic

Figure 2: Question Filter Interface Sketch

3.2 Conceptual Model Design

The conceptual model of the IQB Programming Question Bank Website represents the structure and interactions between the main system components that work together to provide an effective and organized learning environment for programming interview preparation. The main components of this system include registration/login, question browsing and filtering, question detail page, bookmarking system, study calendar, and a comment section.

Based on Figure 3, the registration/login module uses an interaction type of 'command', where users enter their credentials through a registration or login form. These credentials are processed by the backend (SpringBoot) and stored in a MySQL database to enable personalized access to user data such activity history.

The question browsing and smart filtering module has an interaction type of 'explore', as it allows users to navigate through a large set of programming questions using filters such as topic, difficulty level, and programming language. The frontend dynamically updates the displayed list based on the user's selected criteria using Vue.js, while the data is fetched via REST APIs from the backend.

The question detail page is one of the key modules in the system. It has an interaction type of 'view', as users can only view the full problem description and the corresponding correct answer. This interface is designed for reviewing and understanding the logic behind each question. Additionally, users can scroll down to access the comment section to engage in discussions or ask for clarification.

The bookmarking feature has an interaction type of 'select', where users can click a bookmark icon to save a question into their personal list for later review. This status is stored in the database and linked to the logged-in user’s account.

The study calendar module is implemented with an interaction type of 'monitor'. It helps users visualize their daily study activities by displaying completed question counts per day. This view is automatically updated based on user activity logs and provides a motivational overview of their learning progress.

The comment section, found at the bottom of each question detail page, has an interaction type of 'respond'. It allows users to post comments or reply to others, enabling peer-to-peer learning and clarification. Each comment is associated with the respective question and stored in the database.

All components were developed using a modern tech stack: the frontend was built with Vue.js, the backend with Java (SpringBoot and MyBatis), and the database uses MySQL. Data transfer between client and server is handled using secure RESTful APIs.

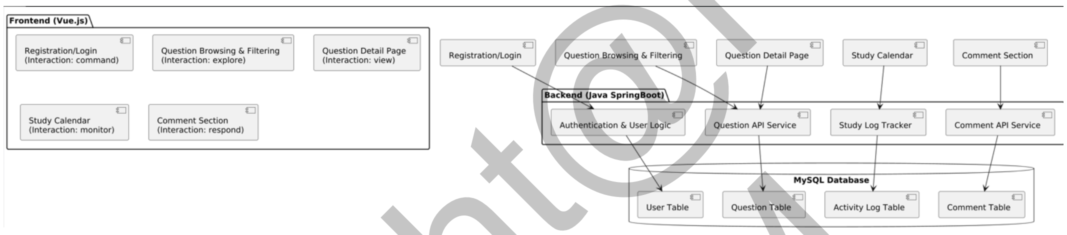

Frontend (Vue.js)
Registration/Login (Interaction: command)
Question Browsing & Filtering (Interaction: explore)
Question Detail Page (Interaction: view)
Study Calendar (Interaction: monitor)
Comment Section (Interaction: respond)
Registration/Login
Question Browsing & Filtering
Question Detail Page
Study Calendar
Comment Section
Backend (Java SpringBoot)
Authentication & User Logic
Question API Service
Study Log Tracker
Comment API Service
MySQL Database
User Table
Question Table
Activity Log Table
Comment Table

Figure 3: Conceptual Model of IQB Programming Question Bank Website Modules and Interactions

4.0 RESULTS

4.1 Application Development

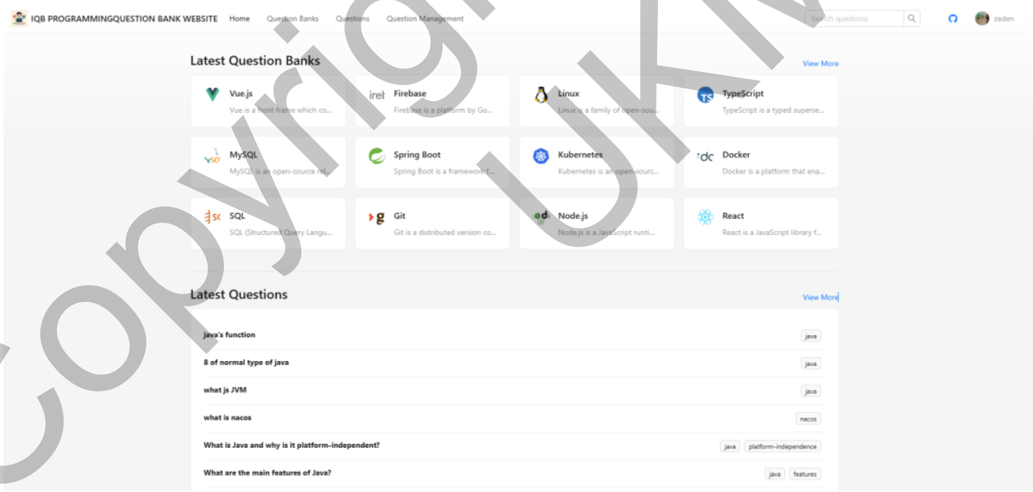
In the development phase, multiple software and frameworks were used to construct the IQB Programming Question Bank Website, starting from interface design to full system deployment. During the interface design phase, Figma was used to design the layout of key pages such as the homepage, question list, and question detail view. These mockups helped define the structure and user flow of the system.

The website was developed using a modern web technology stack. The frontend was built using Vue.js, which provides a reactive and component-based approach to building user interfaces. Page transitions and routing were handled using Vue Router, while Axios was used to establish communication with backend services through RESTful APIs.

For the backend, the system utilized Java SpringBoot, a framework suitable for scalable server-side development. It manages core functionalities such as user authentication, question filtering, answer display, and comment management. RESTful APIs were implemented to support seamless communication between the client and server.

The MySQL database was used to store and manage persistent data, including user accounts, question metadata, full question content and answers, and user-submitted comments. To streamline database operations, MyBatis was integrated into the backend, allowing SQL queries to be mapped to Java objects.

Figure 4 to Figure 8 show the key interfaces of the application. After logging in, users are directed to the Homepage, which provides access to major features such as the question list and study calendar. This interface is shown in Figure 4.



Latest Question Banks

Vue.js
Vue is a front frame which co...

Firebase
Firebase is a platform by Go...

Linux
Linux is a family of open-sou...

TypeScript
TypeScript is a typed superse...

MySQL
MySQL is an open-source rel...

Spring Boot
Spring Boot is a framework f...

Kubernetes
Kubernetes is an open-sourc...

Docker
Docker is a platform that ena...

SQL
SQL (Structured Query Langu...

Git
Git is a distributed version co...

Node.js
Node.js is a JavaScript runti...

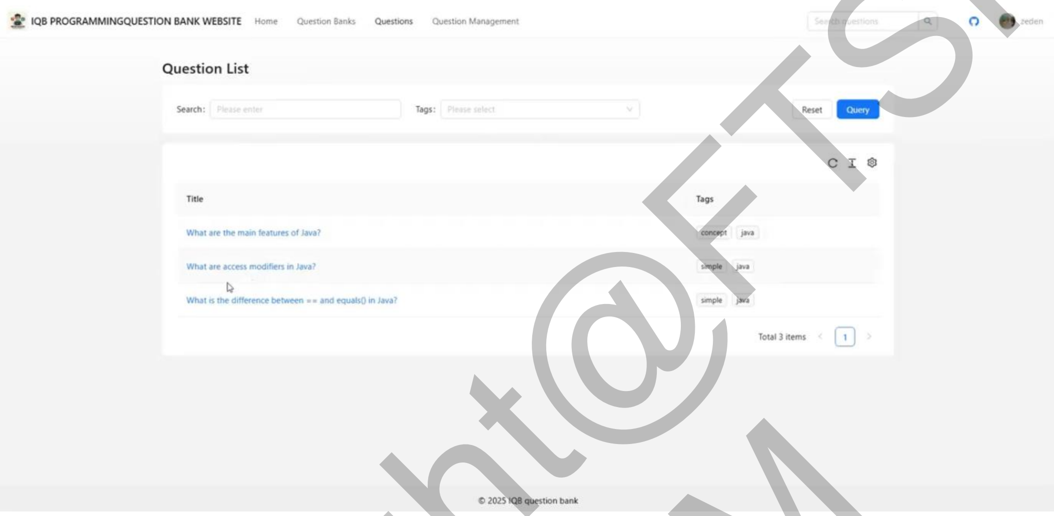
React
React is a JavaScript library f...

Latest Questions

java's function — java
8 of normal type of java — java
what js JVM — java
what is nacos — nacos
What is Java and why is it platform-independent? — java, platform-independence
What are the main features of Java? — java, features

Figure 4: Homepage Interface of IQB Programming Question Bank Website

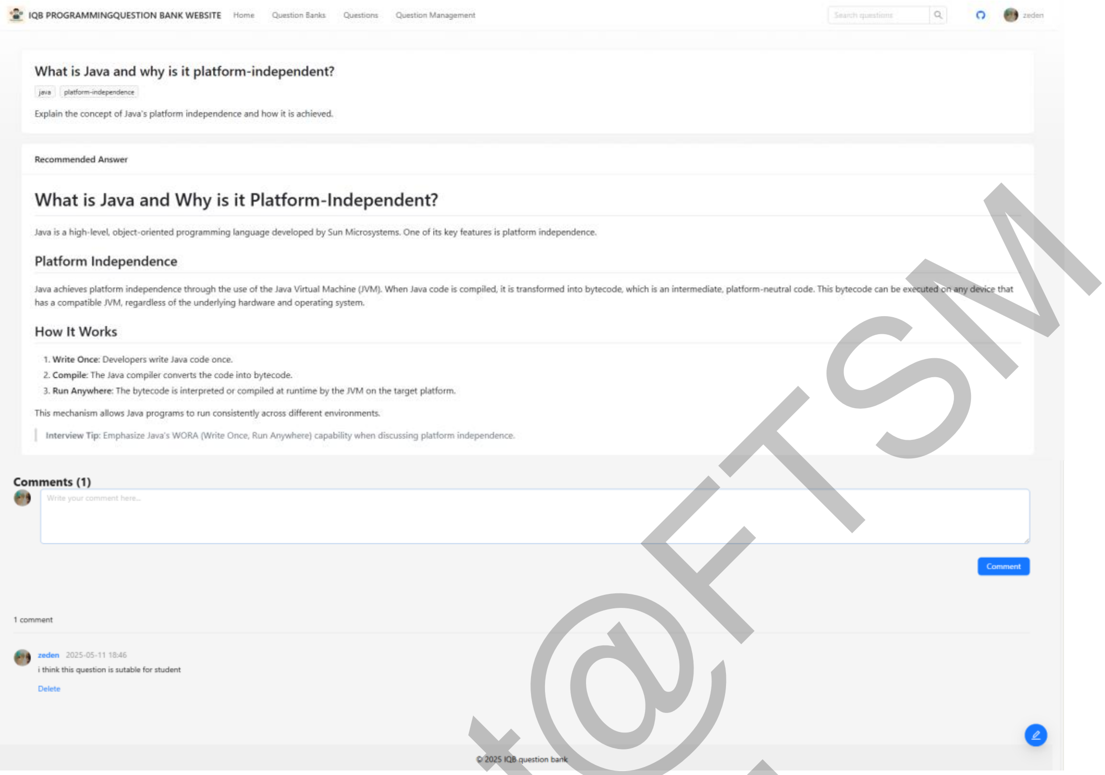
The Question Bank interface presents a full list of programming questions. As shown in Figure 5, users can search and filter questions by programming language, difficulty level, and topic. This enables learners to target specific types of problems during their preparation.



Title	Tags
What are the main features of Java?	concept java
What are access modifiers in Java?	simple java
What is the difference between == and equals() in Java?	simple java

Figure 5: Question Bank Interface with Search and Filter Options

Clicking on a question title opens the Question Detail Page, shown in Figure 6. This interface displays the complete problem description along with the model answer, allowing users to study solution approaches. A comment section is available at the bottom of the page to encourage discussion among users.



What is Java and why is it platform-independent?

java platform-independence

Explain the concept of Java's platform independence and how it is achieved.

Recommended Answer

What is Java and Why is it Platform-Independent?

Java is a high-level, object-oriented programming language developed by Sun Microsystems. One of its key features is platform independence.

Platform Independence

Java achieves platform independence through the use of the Java Virtual Machine (JVM). When Java code is compiled, it is transformed into bytecode, which is an intermediate, platform-neutral code. This bytecode can be executed on any device that has a compatible JVM, regardless of the underlying hardware and operating system.

How It Works

1. **Write Once**: Developers write Java code once.
2. **Compile**: The Java compiler converts the code into bytecode.
3. **Run Anywhere**: The bytecode is interpreted or compiled at runtime by the JVM on the target platform.

This mechanism allows Java programs to run consistently across different environments.

> **Interview Tip**: Emphasize Java's WORA (Write Once, Run Anywhere) capability when discussing platform independence.

Comments (1)

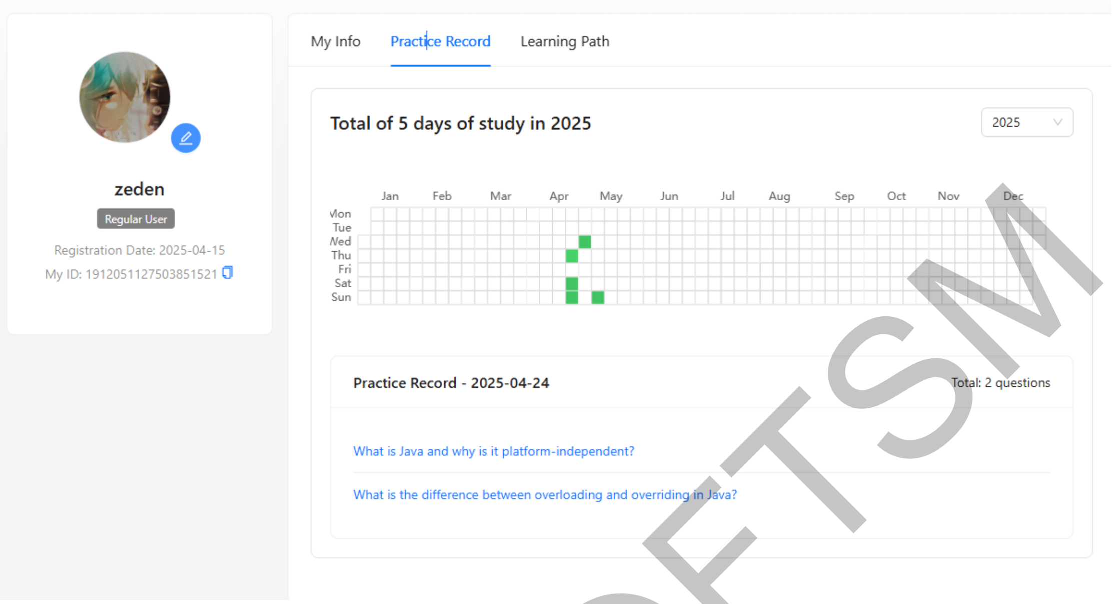
Write your comment here...

1 comment

zeden 2025-05-11 18:46
i think this question is sutable for student
Delete

Figure 6: Question Detail Page with Problem, Answer, and Comment Section

In addition to question review, the platform provides a Study Calendar for users to monitor their activity. As illustrated in Figure 7, the calendar visualizes the number of questions viewed each day, helping users track their consistency and study habits.



zeden

Regular User

Registration Date: 2025-04-15

My ID: 1912051127503851521

Total of 5 days of study in 2025

Practice Record - 2025-04-24

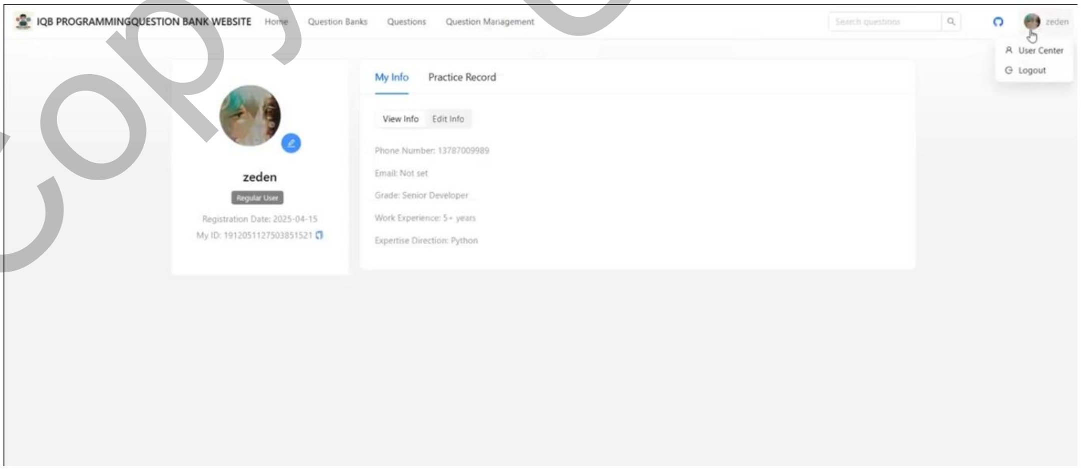
Total: 2 questions

What is Java and why is it platform-independent?

What is the difference between overloading and overriding in Java?

Figure 7: Study Calendar Interface Showing Daily Activity

The system also includes a User Profile interface, where users can update their personal details and access their comment history. This page, shown in Figure 8, supports user information management and offers a centralized space for profile customization.



zeden

Regular User

Registration Date: 2025-04-15

My ID: 1912051127503851521

Phone Number: 13787009989

Email: Not set

Grade: Senior Developer

Work Experience: 5+ years

Expertise Direction: Python

Figure 8: User Profile Interface for Managing Account and Viewing Comments

The IQB Programming Question Bank Website was developed using a modular and responsive architecture to ensure long-term maintainability. All client-server communications are securely handled using REST APIs, and the system is optimized for access on both desktop and mobile platforms.

4.2 Application Evaluation

The evaluation procedure aims to ensure that all functions are working properly. Functional testing aims to identify the resulting errors. In addition, application usability testing was conducted, with participants including undergraduate students from FTSM College. This testing aims to ensure that the application meets the needs of users.

i. Functional Testing

The IQB Programming Question Bank Website underwent a testing phase to evaluate its overall functionality and user experience. Key features such as page navigation, question filtering, question detail display, comment submission, and study calendar tracking were tested to ensure they worked as expected. Black-box testing was applied to verify each module from the user's perspective, focusing on input-output behavior without analyzing the internal code structure. This method allowed for the rapid identification of bugs and ensured that system responses aligned with user actions. All detected issues were promptly addressed during the testing cycle, and the final results confirmed that the website meets its functional requirements and delivers a smooth, user-friendly experience. Functional testing includes six main phases, as shown in Table 1.

Table 1: Tested Functions

Phase	Things	Results
1	Login and authentication	Passed
2	Question list filtering	Passed
3	Question detail and answer display	Passed
4	Comment posting and display	Passed
5	Study calendar activity tracking	Passed

Before functional testing was conducted, the IQB Programming Question Bank Website was deployed on a test server and accessed using multiple web browsers to ensure compatibility across platforms. In Phase 1, the login and authentication functions were tested by registering multiple user accounts and verifying the login process. The system successfully validated credentials, redirected users to the homepage, and prevented unauthorized access.

In Phase 2, the filtering function on the Question Bank page was tested. Users could apply filters such as programming language, and tags to narrow down the question list.

Phase 3 focused on the Question Detail Page. When a question was selected, the system correctly displayed the full problem statement and the official solution. The layout was clear, and all related elements, including syntax formatting, were properly rendered.

In Phase 4, the comment submission and display function was evaluated. Users were able to post comments under each question, and comments were immediately visible in the comment section. Tests confirmed that input validation and timestamp display were functioning correctly.

Finally, Phase 5 involved the Study Calendar. Upon accessing this feature, the system accurately visualized the number of questions reviewed each day. The calendar updated automatically after users viewed or interacted with a new question, providing a real-time study log.

Testing across all five phases showed that the application's key functions operated without any critical errors. Each tested component returned the expected results, confirming that the system is functionally stable and ready for real-world usage.

ii. Usability Testing

Usability testing was conducted to ensure the IQB Programming Question Bank Website meets user expectations in terms of content clarity, interface accessibility, and overall satisfaction. The test aimed to assess whether the system delivers an effective user experience for students preparing for technical programming interviews.

The evaluation was carried out using a Google Form distributed to 25 student participants. They were instructed to browse through the website modules and answer several questions based on their usage experience. The questionnaire focused on four main components: question content, answer explanation quality, user interface friendliness, and overall satisfaction with the website. Rather than using a Likert scale, users provided binary feedback on whether they were satisfied or not with each aspect.

Table 2 summarizes the outcome of the usability test. The results showed that most users were satisfied with all components, with especially high satisfaction in the area of question content and user interface clarity.

Table 2: User Evaluation of the IQB Programming Question Bank Website

Evaluation Criteria	Satisfied	Not Satisfied
Question Content	25	0
Answer Explanation	24	1
User Interface	25	0
Overall Satisfaction	24	1

The feedback suggests that the IQB Programming Question Bank Website provides a clear and useful learning environment, with minimal user dissatisfaction.

Some respondents suggested future enhancements such as better categorization of questions and additional tips for solving problems, which will be considered in future iterations.

5.0 CONCLUSION

The IQB Programming Question Bank Website was developed as an interactive platform to assist students in strengthening their understanding of programming problems, particularly for technical interview preparation. The system integrates well-structured question listings, categorized filtering, detailed explanations, and a responsive user interface to support effective self-paced learning.

The implementation of web technologies such as Vue.js for the frontend and Spring Boot for the backend, along with MySQL and Firebase for data management, ensures smooth performance and real-time interactivity. Through usability testing and feedback collected from participants, the website was found to be easy to navigate, informative, and aligned with the expectations of programming learners.

The ability to access questions by topic or difficulty level, read curated explanations, and engage with peer comments provides users with a rich learning environment. The system also includes supportive features such as a study calendar to track learning consistency and improve motivation.

In conclusion, the IQB Programming Question Bank Website successfully achieves its goal of providing an accessible and focused learning platform for programming students. It serves as a supplementary resource that complements existing study materials, and with further improvements based on user feedback, the system has the potential to support broader academic and interview preparation needs in the future.

6.0 REFERENCES

Spaho, E., Çiço, B., & Shabani, I. J. C. (2025). IoT Integration Approaches into Personalized Online Learning: Systematic Review. *14*(2), 63.