# **Acknowledgement**

I'd want to thank everyone who helped make this project a success. First and foremost, I'd like to express my gratitude to my supervisor, [Supervisor's Name], for his invaluable guidance and continuous assistance.  Their advice and support helped shape the direction and quality of this quiz application. I'd also like to thank my friends  whose recommendations and support contributed significantly to the project's success. Their eagerness to share ideas and provide feedback helped to improve the application's functionality and design. In addition, I am grateful towards National College of Computer Studies for giving me the environment and resources I needed to do this job. The academic environment and the availability of research materials have made a huge difference in the caliber of this work.

# **Abstract**

This report describes the design and development of a Java-based quiz application targeted at testing users' understanding of Java programming fundamentals. The application includes a user-friendly graphical interface built using Swing that allows users to take a quiz, view their results, and retake it as needed. The application tracks the user's score and displays it at the conclusion, along with the option to play again. The system also features a login interface for customizing the experience. The report explains the application's architectural design, essential features, and functionalities, with a focus on its structure, user interface, and fundamental programming logic. This project demonstrates how to use Java programming techniques and object-oriented principles to create interactive software solutions

# **Introduction**

## **Organization Description**

This project is being built for MeroSiksha, an organization in Kathmandu, Nepal that specializes in offering educational materials and technologies to improve professional and student learning experiences. The company specializes in developing interactive platforms that increase learning's effectiveness, accessibility, and engagement for a broad range of users. Its services cover a wide range of topics, such as professional certifications, programming languages, and academic disciplines.

The quiz app that is being built will be a seamless integration with the company's current offerings. Its multiple-choice questions are intended to assist users in honing their skills in programming. Because the questions and answers are randomly generated, the technology keeps the information interesting and provides a dynamic experience for every user.

## **Objectives**

* Develop a Java-based quiz application to provide an interactive and engaging learning experience.
* Provide a user-friendly platform for students to test their knowledge.
* Supportbetterunderstanding of concepts and facilitate exam preparation

## **Scope and Limitation**

### **Scope**

This project's scope includes creating an intuitive Java quiz application with a number of multiple-choice questions about Java programming. It will track user scores, provide replay functionality for better performance, and randomly select questions for a distinctive quiz experience. Important Java subjects including data types, exceptions, collections, and object-oriented programming will all be covered in the test. By letting users add their names, the program will provide a personalized experience. It will be made to function well on desktop platforms, emphasizing front-end development over more complex capabilities like database integration.

### **Limitations**

* **No Database Integration:** Neither user information nor test results are saved in a database by the quiz application.
* **Limited Customization**: Question categories and difficulty levels cannot be customized in the application.Restricted Question Set: The variety of quizzes is limited because the amount of questions is fixed.
* **No User Authentication:** There are no account management features in the application.
* **Platform Dependency:** The program is not compatible with mobile platforms and is only intended to operate in desktop settings.
* **No Feedback or Explanation:** The system doesn't explain why a response is right or wrong.

## **Methodology**

### **Project Framework**

The project is organized into many stages to enable smooth execution and logical progression as follows:

* **Requirement Analysis:** First of all, I identified the quiz application's major features, user demands, and goals.
* **System Design:** In the next step, I laid out the framework of the application, including components such as the user interface.
* **Implementation:** Now, I translated the design into code by breaking it down into smaller parts like the question and answers bank, login, rules, main quiz section, score calculator, and randomization mechanism.
* **Testing:** Then, I performed functional and usability tests to guarantee that the program is bug-free and user-friendly.
* **Deployment:** Finally, I compiled it into an executable format and made it ready for use.

### **Data collection**

In order to construct the quiz application, data collection includes:

* **Survey:** To collect a diverse set of Java-related questions suited for a multiple-choice format, I performed research on educational websites, textbooks, and online question banks.
* **User Feedback:** Early users of the application provided feedback that helped to improve the question bank, interface design, and overall operation.

### **Tools used**

The following resources were utilized in the development of the quiz application:

* **Programming Language:** Java is the programming language used for application development and core logic.
* **Integrated Development Environment (IDE):** Java code was written and compiled using Apache Netbeans.
* **User Interface:** The graphical user interface (GUI) is designed using javax.swing library..

# **Analysis of Activities done and problem solved**

## **Requirement Analysis**

The project's objectives, user requirements, and technical specifications were all carefully outlined throughout the requirement analysis phase. To make sure that the project development is in line with the expectations of the stakeholders and users, this phase is essential. needs were gathered using a variety of techniques and divided into functional and non-functional needs.

### **Requirement Collection Method**

The following techniques were employed in order to gather the requirements for the quiz application project:

* **Interviews:** Insights regarding the features required for the application were obtained through conversations with instructors and prospective users.
* **Surveys:** To learn about the expectations, preferences, and technical expertise of potential users with regard to quiz applications, surveys were sent to them.
* **Research:** A benchmark for functional and non-functional needs was established by looking at academic materials, Java programming courses, and existing quiz applications.
* **Use Cases:** To make sure the project will satisfy user expectations, use case analysis was carried out to model the interactions between users and the system.

### **Functional Requirement**

The core functionalities of this project are as follows:

* **User Registration:** In order to monitor their progress, users should be able to register by entering their name.
* **Dynamic Question Display:** To give every user a different experience, the system should show questions at random.
* **Multiple-Choice Questions (MCQs):** There will be one right response for every question, with four possible answers.
* **Score tracking:** Users ought to be able to view their total score.
* **Timer:** To make the quiz more challenging, a timer should be set for each questions.

### **Non-functional Requirement**

The requirements for smooth functioning of the system are as follows:

* **Reliability:** The system must function well, producing correct questions and answers without crashing.
* **Usability:** The system should have a simple user interface that allows for easy navigation.
* **Speed:** The system must respond quickly, especially when loading questions and displaying results. The application should load in a few seconds, minimizing wait time.
* **Maintainability:** New features should be easy to add, while current ones should be simple to maintain or modify.

## **Feasibility Study**

The feasibility study assesses the viability of building and implementing the quiz application by considering a variety of factors such as technical, operational, economic, and scheduling feasibility. This guarantees that the project is manageable and will benefit both users and stakeholders.

### **Technical Feasibility**

The main considerations include:

* **Current Resources:** The system will be built on Java, a programming language that can run on any basic computer with low hardware requirements. The system will require:
  + A computer or server with a basic processor (1.5 GHz or faster).
  + At least 2GB of RAM.
  + The Java Development Kit (JDK) plus any basic Integrated Development Environment (IDE), such Eclipse or Apache Netbeans.
* **Existing Technology:** The technology needed to build and run the system is readily available, and it may use existing libraries to handle the features making the development process easier.
* **Technical Skills:** The project necessitates basic to intermediate understanding of Java programming, including object-oriented principles and user interface design.

### **Operational Feasibility**

Operational feasibility determines if the proposed system can meet the expected user needs and is simple to run:

* **Meeting the Requirements:** The system's fundamental functionality, such as user login, quiz questions, randomization, and scoring, will be tailored to the operational demands of educational institutions and other users who require a quiz platform. The functionality complies with the requirements identified during the analysis process.
* **Ease of Use:** The system is intended to be simple, straightforward, and user-friendly. An easy-to-use interface will be created to ensure smooth interaction with a short learning curve.
* **Maintenance:** The system will be built with simple code, making it simple to maintain and update as needed.

### **Economic Feasibility**

Economic feasibility assesses whether the project's benefits justify the expenditures associated with its development and operation.

* **System Design:** The main costs will be for software development (developer hours), testing, and user interface design. These costs will be kept low because to the usage of open-source technologies and available development resources.
* **Hardware and software:** The essential hardware (computers or servers) and software are inexpensive or free, lowering overall costs.
* **Operating Costs**: Because the project does not rely on server hosting or databases, the running costs are mostly focused on the resources and tools needed to develop and maintain the quiz application, such as personal computing devices.

### **Schedule Feasibility**

Schedule feasibility evaluates the time required to complete the project and if the development will meet deadlines.

* **Deadline for development:** The project is anticipated to be completed in four stages over a period of 1 month.
  + **Phase 1**: Requirement analysis and system design.
  + **Phase 2:** Development of basic features, including user the quiz module and user login.
  + **Phase 3:** Integration and Testing.
  + **Phase 4:** Deployment.
* **Feasibility:** The timeline is possible given the resources available. Development and testing can proceed as planned, with plenty of time to make any necessary changes or improvements based on feedback.

## **Module Description**

## **Testing**