

ONLINE QUIZ APPLICATION

INTERDISIPILINARY PROJECT

Submitted in partial fulfilment of the requirements for the award of Bachelor
of Technology degree in Information Technology

by

KOTERU AMRUTHA (Reg.No-41120097)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY-1 UNIVERSITY BY UGC

Accredited with Grade “A++” by NAAC

12B Status by UGC |Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this internship Report is the Bonafede work of **Koteru Amrutha (41120097)** who carried out the project entitled "**ONLINE QUIZ APPLICATION**" under my supervision from January 2024 to May 2024.

Internal Guide

Ms. P.Shamili MTech.,

Head of the Department

Dr. R Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on

8-5-24

Internal Examiner

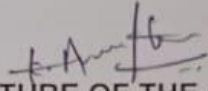
External Examiner

DECLARATION

I, Koteru Amrutha (41120097) hereby declare that the Project internship Report titled **ONLINE QUIZ APPLICATION** done by me at HCL, is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 8-5-24

PLACE: CHENNAI


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **DR. T. Sasikala M.E., PH.D., Dean**, School of Computing, **Dr R Subhashini M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. P.Shamili MTech.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

The Java platform of a quiz application that enables students to engage in quiz activities. Based on their performance in the quizzes, students will receive scores that are visible to students. The application incorporates an authentication process, allowing students to log in using their designated credentials. There will be login window in the students should enter the name after entering the name the student can see the two buttons one is rules button and other is back button if choose rules button, he can see the rules in rules window. Now the student can read the rules and click next to write the quiz the student can see 10 questions he will enter every question and he will click next after the 10th question he will submit and he will get the score in score window. The provided application serves the purpose of conducting quizzes, catering to students. Students are empowered to choose the quiz category aligned with their performance level, and subsequently, their marks are presented after appropriate evaluations. This access permits them to view students' marks and introduce supplementary questions to the students as deemed necessary. The project employs modern web development technologies and methodologies, including HTML, CSS, JavaScript, and backend frameworks like Django or Flask. The development process follows an iterative approach, incorporating user feedback to iteratively improve the application's usability and functionality.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	iv
	LIST OF FIGURES	v
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 FUNCTIONALITIES	2
2	METHODOLOGY	3
	2.1 AIM	3
	2.2 SCOPE OF PROJECT	7
	2.3 MATERIALS USED	7
	2.4 METHODS USED	9
	2.4.1 USE CASE DIAGRAM	10
	2.4.2 DESIGN OF PROJECT	13
3	RESULTS AND DISCUSSION	18
	3.1 RESULTS	18
	3.2 DISCUSSION	23
4	CONCLUSION AND FUTURE WORK	26
	4.1 CONCLUSION	27
	REFERENCES	
	APPENDICES	
A.	SOURCE CODE	
B.	SCREENSHOTS	

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
1.1	Basic quiz system	2
2.1	Quiz application	5
3.1	Login Authentication	6
4.1	Water fall model	9
5.1	Use case	10
5.3	3-tier architecture	12
6.1	Database	13

CHAPTER-1

INTRODUCTION

1.INTRODUCTION

The Examination application is used for conducting examination for school or college students or this software can also be used by the company for the recruitment process. At first, the student is needed to register his/her name along with all the information needed and need to give the ID of the student for the login process. Using their Id, the student can login into the examination. Then user need to select the Category for which he/she wants to start the examination. Next procedure is answering the examination. As soon as the student selects the examination menu strip, the questions with 4 options will be displayed. The student has to type any one option and click on next option. ON this click next set of question and answers will be displayed. This will continue till the end of the question. The admin has the authority to add or modify any questions or options. The answers of each question will be saved in the database. The result option shows the result of all students. This quiz website provides a great platform for practice and solving query by taking feedback from the user, which is having a great user interface.

The proposed work in this field is available on internet in huge amount like; many websites are available for playing quizzes online and query solving of the users. These websites provide a large variety of questions (Big Domain). This website provides question and answer for quiz, query box for query solving. Also, these websites provide hug amount of study material for the users.

1.2 FUNCTIONALITIES OF THE PROJECT:

- Able the examiners to punch the MCQ questions online;
- Able the users to solve the questions online;
- Examiners can manage the information regarding exam;
- Correct answers will be evaluated by system
- Users can see their result after submitting the test as shown in figure 1.1.

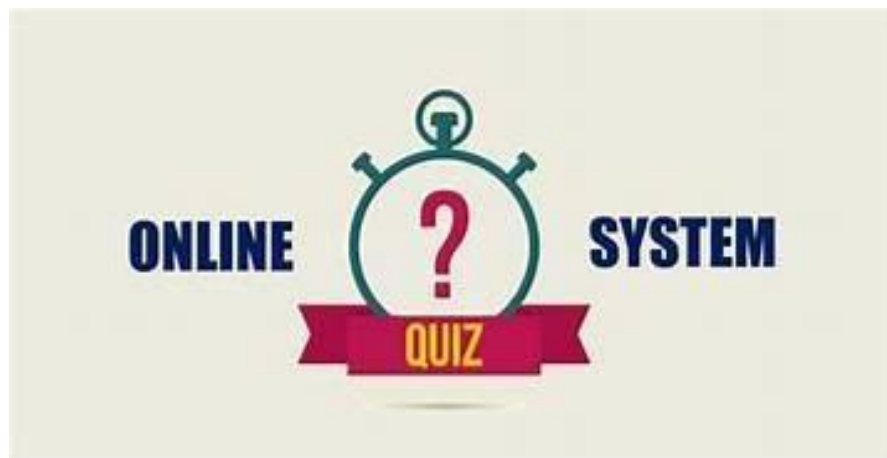


FIGURE.1.1. BASIC QUIZ SYSTEM

CHAPTER -2

METHODOLOGY

2.1 AIM

The aim of this project is to computerized the existing manual system and help the examiners to save their valuable time and important data. Apart from this, data which are exist in this system, will exist for long period of time and will be easily accessible. This project helps the examiners to manage their services in a good way and provide a better service to their users. Our aim is to develop an application for the users in which a user can attempt any number of quiz related to his/her choice”.

- 1.Firstly, we have to make interface for Home Page, Registration, Login Page, Question Attempting Forum, Result Page and profile of user and as well as for Admin. These all pages have connectivity with the server and database.
2. Currently, there are websites which only provide limited number of quizzes related to different domain.
3. Many websites do not have a single platform for quizzes related to Technical, G.K, Aptitude, Grammar, etc. And there is not a website where the users can upload his/her question and answer for others.
4. We have to develop an application which can resolve all the problems.

2.2 SCOPE OF THE PROJECT

The provided application serves the purpose of conducting quizzes, catering to both students and teachers with distinct roles. Students are empowered to choose the quiz category aligned with their performance level, and subsequently, their marks are presented after appropriate evaluations. Teachers, on the other hand, access the application through separate login credentials. This access permits them to view students' marks and introduce supplementary questions to the students as deemed necessary. The teacher interface provides a window wherein student marks are showcased, and where the option to append extra questions to specific students is available. Scope of this project is very broad in terms of other manually checking yourself and few of them are: -

- This can be used in educational institutions as well as in corporate world.
- Can be used anywhere any time as it is a web-based application (user Location doesn't matter).
- No restriction that examiner has to be present when the candidate takes the test.

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Online Quiz System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly. Our project aims at Business process automation, Le. we have tried to computerize various processes of Online Quiz System.

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.

- It satisfies the user requirement
- Be easy to understand by the user and operator
- Be easy to operate
- Be easy to operate Have a good user interface
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget as shown in figure 2.1.

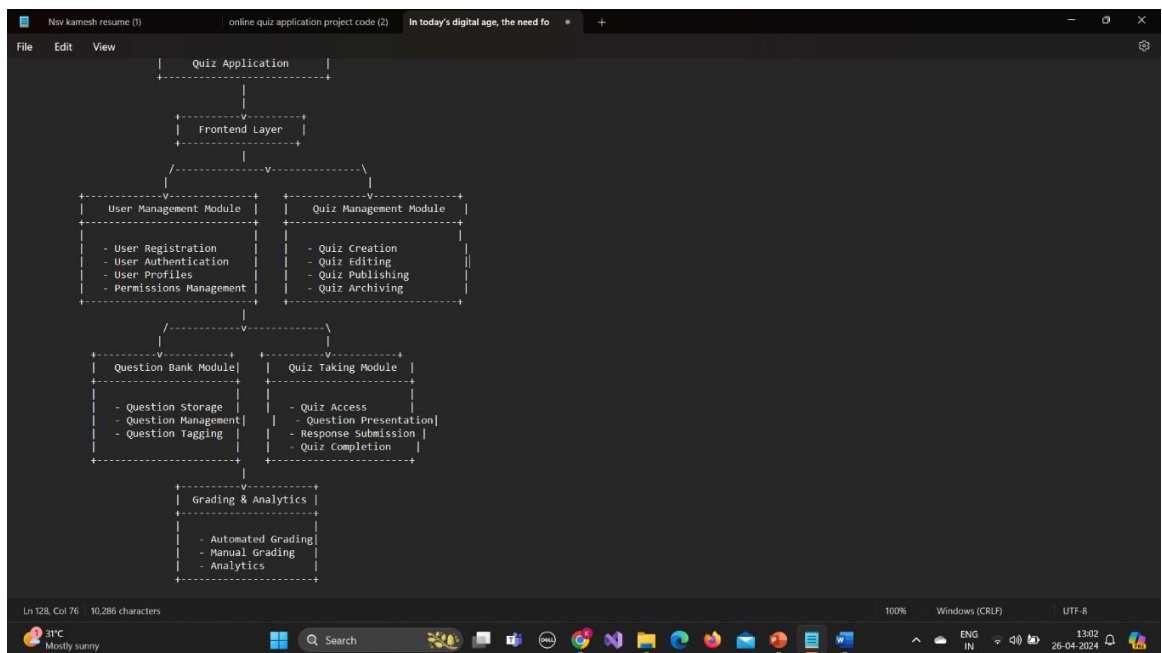


FIGURE:2.1. BASIC DIAGRAM FOR QUIZ APPLICATION

Student --> login select the
subject, answer to the question

marks have to be displayed,

limit the marks for pass and fail,

Optional: Grade will be added as per the marks awarded

Teacher: --> login with the different credentials,

Check the Marks

If there is anything needs to be updated in the questions can modify,

extra questions can also be added by the Teacher as shown in figure 3.1.

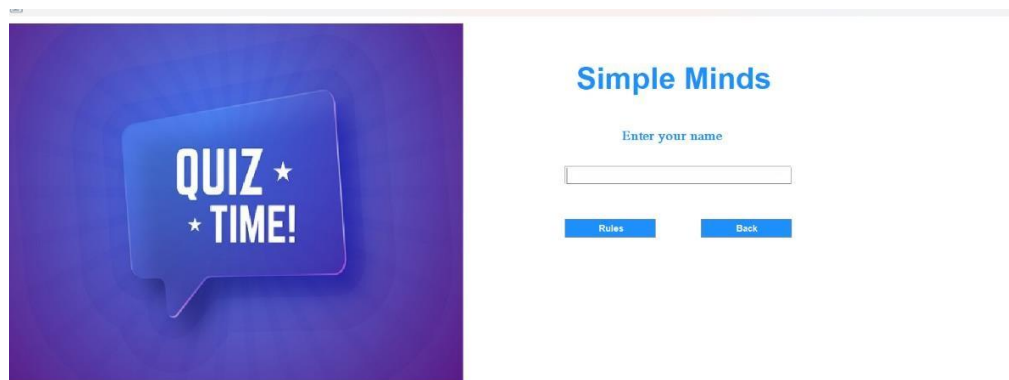


FIGURE:3.1 LOGIN AUTHENTICATION

2.3 MATERIALS USED

Randomly select questions from a question bank. Ensure a balanced mix.

Quiz Configuration:

- Allow users to set parameters (number of questions, time limit, etc.).
- Implement an algorithm to organize questions based on user.

User Authentication:

- Use secure authentication algorithms to manage user accounts.

Randomization:

- Shuffle questions to prevent a fixed sequence in each quiz attempt.
- Randomize answer choices to avoid patterns.

Scoring:

- Implement an algorithm to calculate scores based on correct/incorrect.
- Consider penalties for unanswered questions or incorrect responses.

Adaptive Difficulty:

- Adjust difficulty based on user performance in previous quizzes.
- Implement algorithms that adapt to the user's skills

Time Management:

- Incorporate a timer algorithm to monitor and enforce time limits.

Data Storage:

- Use a database system to store questions, user profiles, and quiz.
- Optimize queries for efficient data retrieval.

Security:

- Employ encryption algorithms to secure user data.
- Implement measures to prevent cheating, such as limiting question.

Feedback Generation:

- Provide detailed feedback on quiz performance.
- Use algorithms to generate insightful comments based on answers.

Reporting:

- Generate reports on user performance, quiz statistics, and trends.
- Utilize data analytics algorithms for meaningful insights.
- User Interface (UI)

2.4 METHODS USED

The programming language used for the development of the project is JAVA and the software model used is the classic lifecycle model.

Waterfall Process model:

The Classical Life Cycle or waterfall Process Model was the first process model to present a sequential framework, describing basic stages that are mandatory for a successful software development model. Requirement analysis, design, coding, testing, and maintenance. Once a phase is done, it is not repeated again that is movement in to meet I case of large projects. It formed the basis for most software development standards and consists of the following phases: Requirement analysis, design, coding, testing, and maintenance as shown in life cycle.

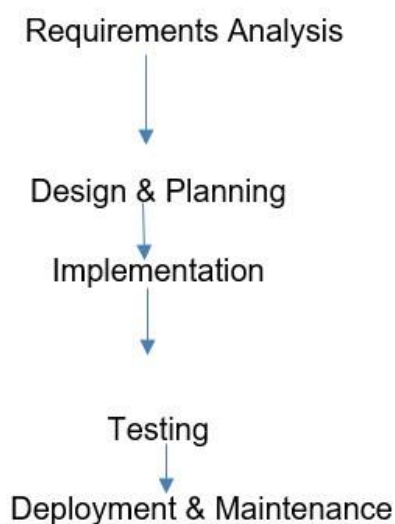


FIGURE:4.1 WATERFALL LIFE CYCLE

2.4.1 USECASE DIAGRAM

A use case diagram at its simplest is a representation of user's interaction with the system that shows the relationship between the user and the different use case in which the users is involved. A use case diagram can identify the different types of users of a system and the different use case.

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Online Quiz System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly. Our project aims at Business process automation, Le. we have tried to computerize various processes of Online Quiz System.

The Classical Life Cycle or waterfall Process Model was the first process model to present a sequential framework, describing basic stages that are mandatory for a successful software development model. Apart from the usual advantages of modular software with well-define interface, the Three- tier architecture is intended to allow any of the three tiers to be upgraded or replace independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the use of interface code. Typically, the user interface runs on a desktop pe or workstations and uses a standard graphical user interface functional process logic that may consist of one or more separate modules running on workstations on application server, and RDBMS on a database server or mainframe that contains the computer data storage logic.

TABLE 1: SPECIFICATION

NAME OF THE USECASE	LOG-IN
Actor	User/student
Precondition	None
Primary flow of event	Email ePassport
Alternate flow of event	If the user or password is wrong shows message "incorrect email id and password"
Post condition	If successful log and come to homepage
Use case termination	cancel

Requirement analysis, design, coding, testing, and maintenance. Once a phase is done, it is not repeated again that is movement in the waterfall goes one to the next and the vice versa is not supported, deadlines are difficult to meet in case of large projects. The Classical Life Cycle or waterfall Process Model was the first process model to present a sequential framework, describing basic stages that are mandatory for a successful software development model. Apart from the usual advantages of modular software with well-defined interface, the Three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the use of interface code. Three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology as shown in table 1.

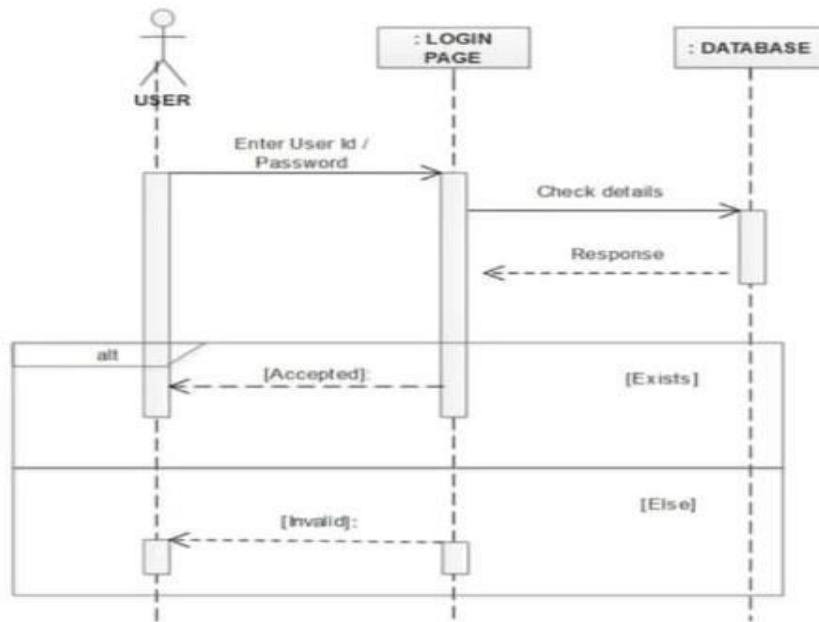


FIGURE.5.2 SEQUENCE DIAGRAM FOR LOGIN

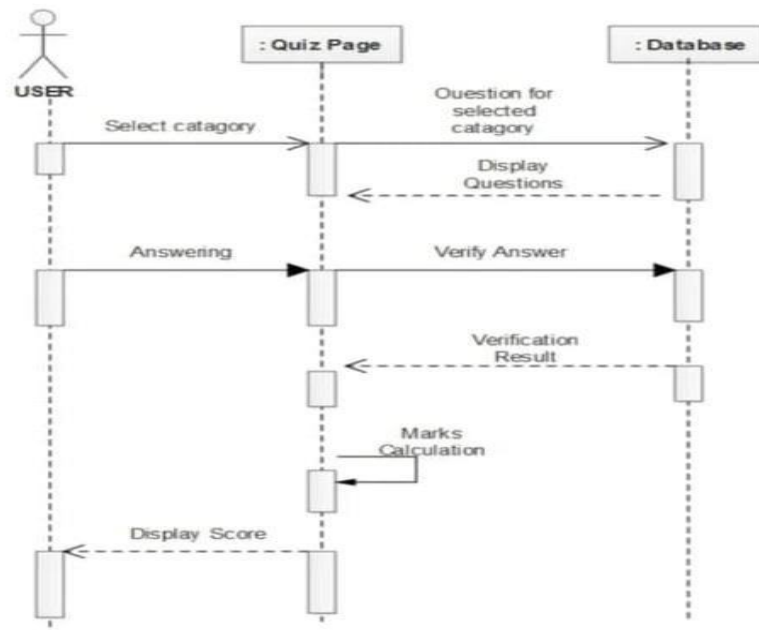


FIGURE.5.3 SEQUENCE DIAGRAM FOR PLAY QUIZ

2.4.2 DESIGN

Three-tier architecture is a client-server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platform. Apart from the usual advantages of modular software with well-defined interface, the Three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the use of interface code. Typically, the user interface runs on a desktop pc or workstations and uses a standard graphical user interface functional process logic that may consist of one or more separate modules running on workstations on application server, and RDBMS on a database server or mainframe that contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture") as shown in figure 5.4.

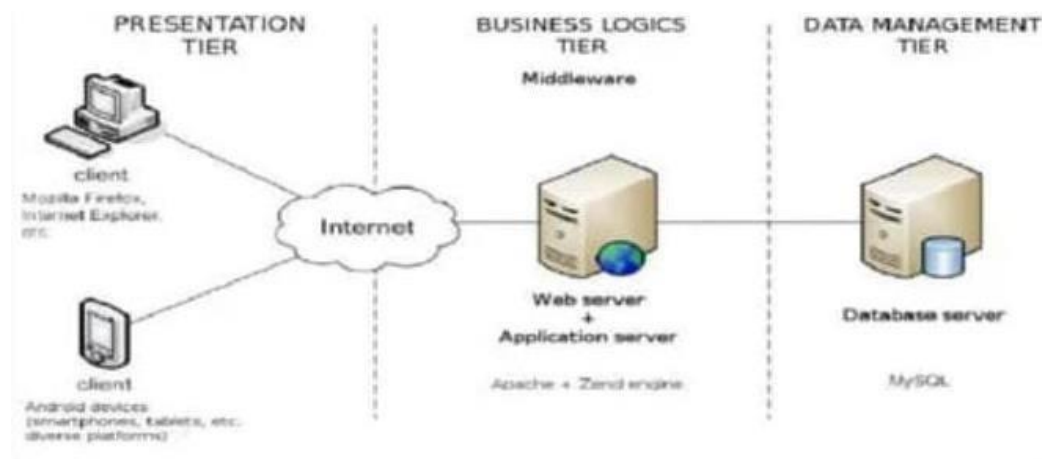


FIGURE:5.4-TIER ARCHITETURE DIAGRAM

2.5 DATABASE DESIGN

Database normalization is a technique of organizing the data in the database. Normalization is systematic approach decomposing table to eliminate data redundancy and undesirable characteristics like insertion, update and deletion anomalies. It is a multi-state process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes:

- Eliminating redundant data.
- Ensuring data dependence makes sense i.e. data is logically stored.

Our table in the database is in 1 NF form. A table is said to be in 1NF if both the following conditions hold. A relation will be 1 NF if it contains an atomic value. It states that an attribute of a table cannot hold multiple values. It must hold single valued attributes. 1 NF disallows the multi-valued attribute, composite attribute, and their combinations as shown in table 2.

TABLE 2: DATABASE TABLE

Name
Password
Email id
Mobile number
Address
Dob

2.6 ALGORITHMS USED

- Implement algorithms for an intuitive and user-friendly interface.
- Ensure responsiveness and adaptability across various devices.

Creating an online quiz application involves various algorithms for different.

User Authentication:

- Algorithm for user registration and login.
- Password hashing and salt generation.

Quiz Creation:

- Algorithm to create quizzes with questions and answers.
- Managing quiz metadata like title, description, and time limit.

Quiz Storage:

- Storing quizzes and questions in a database.
- Designing a schema for efficient data retrieval.

User Interface:

- Design algorithms for presenting quizzes and questions to users.
- Implementing user-friendly navigation.

Quiz Taking:

- Algorithm for presenting questions one-by-one to users.
- Timer management to enforce time limits.

Scoring:

- Algorithm to calculate and store user scores.

- Handling different scoring methods (e.g., points per correct answer).

Leaderboard:

- Algorithm to rank users based on their scores.
- Managing ties and displaying leaderboard data.

Question Randomization:

- Randomly selecting questions from a quiz.
- Ensuring fair distribution of questions.

Security:

- Algorithms for preventing cheating (e.g., randomizing answer order).
- Protection against SQL injection and other security threats.

Result Calculation:

- Algorithm for calculating the final quiz score.
- Handling partial credit for partially correct answers.

User Feedback:

- Algorithm for providing feedback on quiz performance.
- Storing and displaying correct answers after the quiz.

Database Management:

- Algorithms for CRUD operations on quizzes, questions, and user data.

Notifications:

- Algorithm for sending notifications (e.g., email reminders for upcoming).

User Progress Tracking:

- Tracking user progress in quizzes and displaying it.

Performance Optimization:

- Algorithms to optimize database queries and application performance.

Error Handling:

- Implementing error-handling algorithms for graceful degradation.

User Analytics:

- Algorithm to collect and analyse user behaviour data for improvement.

Scalability:

- Algorithms for scaling the application as the user base grows.

Accessibility:

- Algorithms for ensuring the application is accessible to all users.

Testing and Debugging:

- Algorithms for testing the application, identifying bugs, and fixing

CHAPTER-3

RESULTS AND DISCUSSION

3.1 RESULTS

CODE FOR ONLINE QUESTION APPLICATION:

The project contains four classes:

- Login
- Rules
- Quiz
- Score

Login Functionality:

- Users can create an account or log in with existing credentials.
- The application verifies the credentials and grants access to the quiz.

Adding Credentials:

- Users can sign up by providing necessary information such as username, email, and password.
- The application should ensure secure storage of user credentials, preferably by hashing passwords and storing them in a database. Randomly shuffle the order of questions to prevent predictability and ensure fairness. Implement algorithms for smooth navigation between questions, allowing users to move forward, backward, or skip questions.

Scoring:

- For each correct answer, the user earns points based on the difficulty level of the question.
- Incorrect answers may result in a deduction of points or no points awarded.
- The application keeps track of the total score as the user progresses through the quiz.

Result:

- After completing the quiz, the application displays the user's total score.
- Users can see which questions they answered correctly and incorrectly.
- Optionally, the application can provide feedback on incorrect answers, such as the correct answer and an explanation.
- The online quiz application project aimed to provide a seamless and engaging platform for users to participate in quizzes remotely. Developed with user-friendliness in mind, the application offered a diverse range of quizzes catering to various interests and age groups. Through intuitive design and robust functionality, users could easily navigate, select, and participate in quizzes of their choice. The project emphasized scalability and security, ensuring smooth performance even during peak usage times and safeguarding user data.

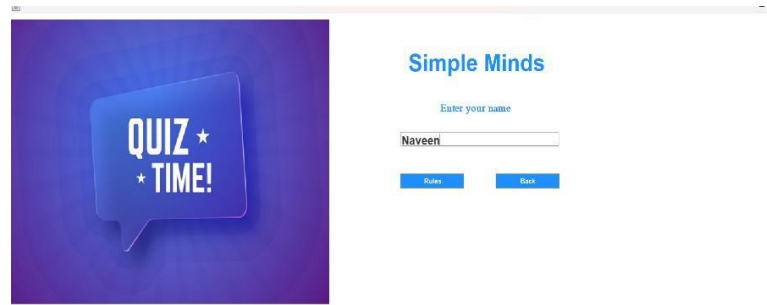


FIGURE:5.5 LOGIN IMAGE

To access the features of our online quiz application, users need to create an account by adding their login credentials. Upon visiting the application's website, users are prompted to register by providing essential information such as a unique username, a valid email address, and a secure password. Once the user submits this information, our application securely stores their credentials in our database, utilizing encryption techniques to safeguard sensitive data. For returning users, the login process involves entering their This ensures that only authorized individuals can access the quizzes and maintain the integrity of their progress and scores within the application as shown in figure 5.5.

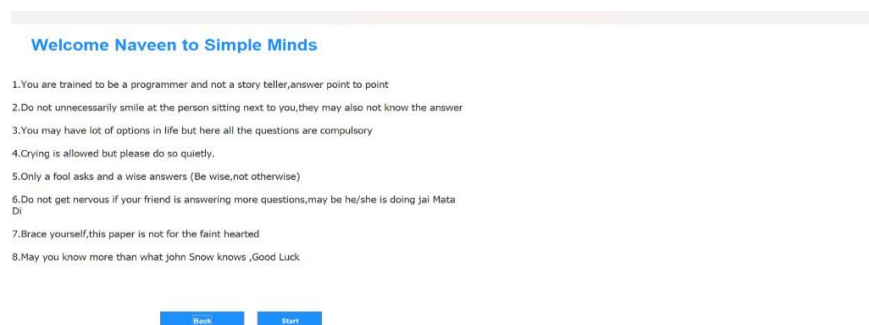


FIGURE:5.6 INSTRUCTIONS

Welcome to our online quiz platform! To begin, simply log in using your registered credentials or sign up if you're new here. Once logged in, you'll find a variety of quiz categories to choose from. Select a category that interests you and start the quiz. You'll be presented with a series of questions, each with multiple-choice answers or open-ended input fields. Take your time to read each question carefully and select the answer you believe is correct. Be mindful of any time limits for each question. After completing the quiz, you'll receive your total score and be able to review your answers. Feel free to explore different quiz categories and challenge yourself. Happy quizzing as shown in figure 5.6.

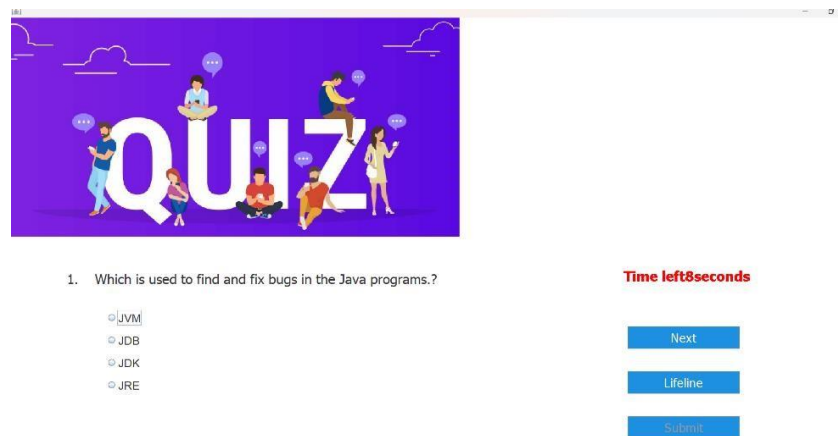


FIGURE:5.7: QUESTIONS

our online quiz application, multiple-choice questions (MCQs) are an integral part of the quiz-taking experience. When you begin a quiz, you'll encounter a series of MCQs, each presenting a question along with several answer options. Your task is to carefully read the question and select the most appropriate answer from the provided choices. Take your time to consider each option before making your selection. Remember, only one answer is correct for each question, so choose wisely. Once you've made your choice, simply click on the corresponding option to submit your answer as shown figure 5.7.

This format offers several advantages, including efficient assessment of knowledge and understanding across a wide range of topics, ease of administration, and objective grading. Additionally, MCQs can assess higher-order thinking skills such as analysis, evaluation, and application when designed appropriately. The process of crafting effective MCQs involves careful consideration of question clarity, distractor quality, and alignment with learning objectives. Moreover, incorporating varied question types, such as single-best answer, multiple-correct answer, and scenario-based MCQs, can enhance the depth and validity of assessment. Overall, MCQs serve as a valuable tool for evaluating learner comprehension, promoting active learning, and providing constructive feedback to both educators and learners.

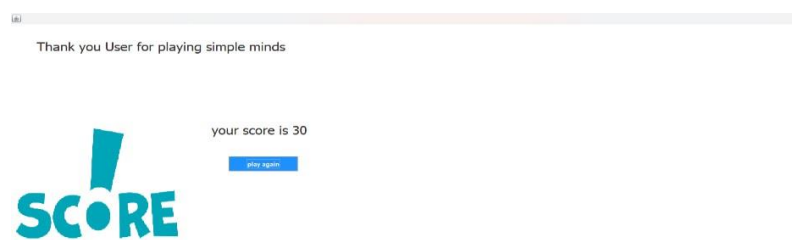


FIGURE:5.8: SCORE IMAGE

In our online quiz application, scoring is an essential aspect of assessing your performance. As you progress through the quiz, each correct answer earns you points based on the difficulty level of the question. Conversely, incorrect answers may result in a deduction of points or no points awarded, depending on the quiz's scoring system. Our application keeps track of your total score in real-time, so you can see how you're doing as you go along. At the end of the quiz, your total score is calculated based on the sum of points earned for each correct answer. You'll receive immediate feedback on your performance as shown in figure 5.8.

Our application keeps track of your total score in real-time, so you can see how you're doing as you go along. At the end of the quiz, your total score is calculated based on the sum of points earned for each correct answer. When you begin a quiz, you'll encounter a series of MCQs, each presenting a question along with several answer options. Your task is to carefully read the question and select the most appropriate answer from the provided choices. Take your time to consider each option before making your selection. When you begin a quiz, you'll encounter a series of MCQs, each presenting a question along with several answer options.

3.2 DISCUSSIONS:

Creating a discussion feature in an online quiz application using Java involves several steps:

Database Setup:

- Choose a database system (e.g., MySQL, PostgreSQL) to store discussion
- user ID, quiz ID, content, timestamp, etc.

Backend Development:

- Set up a Java backend server using a framework like Spring Boot.
- Create API endpoints for discussions, including functionalities like creating, reading, updating, and deleting posts.
- Implement authentication and authorization to ensure that only authorized users can access the discussion features.

User Interface:

- Develop a user interface for discussions using Java Swing, JavaFX, or a web framework like Spring MVC.
- Display discussion threads, posts, and allow users to interact.

Integration with Quiz:

- Integrate the discussion feature into your quiz application so that users can access discussions related to specific quizzes.
- Link discussions to quizzes using unique identifiers (e.g., quiz IDs).
- Implement real-time updates using technologies like WebSocket or server-sent users with live discussions without refreshing the page

Security:

- Implement security measures to prevent common vulnerabilities like SQL injection and Cross-Site Scripting (XSS).
- Ensure user data is securely stored and transmitted.

Testing:

- Thoroughly test the discussion feature to ensure it works correctly and is free of bugs. A quiz application serves as a multifaceted tool for engagement, learning, and assessment across a spectrum of contexts. Through its intuitive user interface and seamless functionality, it offers a platform where users can immerse themselves in interactive quizzes, assessments, and learning activities. At its core, the application revolves around user-centric design principles.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points: -

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of Purpose. Scopes, and applicability.
- We define the problem on which we are working in the project
- We describe the requirement Specifications can be done on these things of the system and the actions that
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.
- Finally, the system is implemented and tested according to test cases.

4.2 FUTURE SCOPE OF PROJECT

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

1. We can add printer in future.
2. We can give more advance software for Online Quiz System including more facilities
3. We will host the platform on online servers to make it accessible worldwide
4. Integrate multiple load balancers to distribute the loads of the system
5. Create the master and slave database structure to reduce the overload of the database queries
6. Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

The above-mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Course and Question. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the Online Quiz System. Enhancements can be done to maintain all the Course, Question, Topic, Student, Quiz.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process. Conversely, incorrect answers may result in a deduction of points or no points awarded, depending on the quiz's scoring system.

REFERENCES

1. Gracia, Maria, "ONLINE APPLICATION DEVELOPMENT" Publisher: IEEE, 2023. DOI: 10.1109/ACCESS.2020.2987523.
2. Johnson, Robert; Brown, Emily, Jake, Harry Paul, Albert Ross "Development and Implementation of Car Care Apps with Flutter: A Case Study", Publisher: IEEE, 2022 DOI: 10.1109/ICTC52510.2021.9620922
3. Patel, Rajesh "Flutter for Automotive Applications", Publisher: IEEE, 2023 DOI: 10.1109/CSNT51715.2021.9509707
4. Lee, Janson "Flutter Framework for Car Care Apps: A Comparative Analysis", Publisher: IEEE, 2024 7111. DOI: 10.1109/WIECON-ECE52138.2020.9397999
- 5 J. Smith, A. Jhonson, "COMPARITIVE ANALYSIS OF ONLINE APPLICATION DEVELPOMENT". J. Documentation processing technol., vol.5, no. 3, pp. 112-125, May 2020

APPENDICES

A.SAMPLE CODE:

```
package quiz.application;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Login extends JFrame implements ActionListener{

    JButton rules, back;
    JTextField tfname;

    Login() {
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getResource("icons/login.jpeg"));
        JLabel image = new JLabel(i1);
        image.setBounds(0, 0, 600, 500);
        add(image);

        JLabel heading = new JLabel("Simple Minds");
        heading.setBounds(750, 60, 300, 45);
        heading.setFont(new Font("Viner Hand ITC", Font.BOLD, 40));
        heading.setForeground(new Color(30, 144, 254));
        add(heading);

        JLabel name = new JLabel("Enter your name");
        name.setBounds(810, 150, 300, 20);
        name.setFont(new Font("Mongolian Baiti", Font.BOLD, 18));
        name.setForeground(new Color(30, 144, 254));
        add(name);

        tfname = new JTextField();
        tfname.setBounds(735, 200, 300, 25);
        tfname.setFont(new Font("Times New Roman", Font.BOLD, 20));
        add(tfname);

        rules = new JButton("Rules");
        rules.setBounds(735, 270, 120, 25);
        rules.setBackground(new Color(30, 144, 254));
        rules.setForeground(Color.WHITE);
        rules.addActionListener(this);
        add(rules);
```

```

        back = new JButton("Back");
        back.setBounds(915, 270, 120, 25);
        back.setBackground(new Color(30, 144, 254));
        back.setForeground(Color.WHITE);
        back.addActionListener(this);
        add(back);

        setSize(1200, 500);
        setLocation(200, 150);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == rules) {
            String name = tfname.getText();
            setVisible(false);
            new Rules(name);
        } else if (ae.getSource() == back) {
            setVisible(false);
        }
    }

    public static void main(String[] args) {
        new Login();
    }
}

package quiz.application;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Quiz extends JFrame implements ActionListener {

    String questions[][] = new String[10][5];
    String answers[][] = new String[10][2];
    String useranswers[][] = new String[10][1];
    JLabel qno, question;
    JRadioButton opt1, opt2, opt3, opt4;
    ButtonGroup groupoptions;
    JButton next, submit, lifeline;

    public static int timer = 15;
    public static int ans_given = 0;
    public static int count = 0;
    public static int score = 0;

```

```

String name;

Quiz(String name) {
    this.name = name;
    setBounds(50, 0, 1440, 850);
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    ImageIcon i1 = new
ImageIcon(ClassLoader.getResource("icons/quiz.jpg"));
    JLabel image = new JLabel(i1);
    image.setBounds(0, 0, 1440, 392);
    add(image);

    qno = new JLabel();
    qno.setBounds(100, 450, 50, 30);
    qno.setFont(new Font("Tahoma", Font.PLAIN, 24));
    add(qno);

    question = new JLabel();
    question.setBounds(150, 450, 900, 30);
    question.setFont(new Font("Tahoma", Font.PLAIN, 24));
    add(question);

    questions[0][0] = "Which is used to find and fix bugs in the Java
programs.?";
    questions[0][1] = "JVM";
    questions[0][2] = "JDB";
    questions[0][3] = "JDK";
    questions[0][4] = "JRE";

    questions[1][0] = "What is the return type of the hashCode() method in the
Object class?";
    questions[1][1] = "int";
    questions[1][2] = "Object";
    questions[1][3] = "long";
    questions[1][4] = "void";

    questions[2][0] = "Which package contains the Random class?";
    questions[2][1] = "java.util package";
    questions[2][2] = "java.lang package";
    questions[2][3] = "java.awt package";
    questions[2][4] = "java.io package";

    questions[3][0] = "An interface with no fields or methods is known as?";
    questions[3][1] = "Runnable Interface";
    questions[3][2] = "Abstract Interface";
    questions[3][3] = "Marker Interface";

```

```

        questions[3][4] = "CharSequence Interface";

        questions[4][0] = "In which memory a String is stored, when we create a
string using new operator?";
        questions[4][1] = "Stack";
        questions[4][2] = "String memory";
        questions[4][3] = "Random storage space";
        questions[4][4] = "Heap memory";

        questions[5][0] = "Which of the following is a marker interface?";
        questions[5][1] = "Runnable interface";
        questions[5][2] = "Remote interface";
        questions[5][3] = "Readable interface";
        questions[5][4] = "Result interface";

        questions[6][0] = "Which keyword is used for accessing the features of a
package?";
        questions[6][1] = "import";
        questions[6][2] = "package";
        questions[6][3] = "extends";
        questions[6][4] = "export";

        questions[7][0] = "In java, jar stands for?";
        questions[7][1] = "Java Archive Runner";
        questions[7][2] = "Java Archive";
        questions[7][3] = "Java Application Resource";
        questions[7][4] = "Java Application Runner";

        questions[8][0] = "Which of the following is a mutable class in java?";
        questions[8][1] = "java.lang.StringBuilder";
        questions[8][2] = "java.lang.Short";
        questions[8][3] = "java.lang.Byte";
        questions[8][4] = "java.lang.String";

        questions[9][0] = "Which of the following option leads to the portability and
security of Java?";
        questions[9][1] = "Bytecode is executed by JVM";
        questions[9][2] = "The applet makes the Java code secure and portable";
        questions[9][3] = "Use of exception handling";
        questions[9][4] = "Dynamic binding between objects";

        answers[0][1] = "JDB";
        answers[1][1] = "int";
        answers[2][1] = "java.util package";
        answers[3][1] = "Marker Interface";
        answers[5][1] = "Remote interface";
        answers[6][1] = "import";
        answers[7][1] = "Java Archive";

```

```

answers[8][1] = "java.lang.StringBuilder";
answers[9][1] = "Bytecode is executed by JVM";

opt1 = new JRadioButton();
opt1.setBounds(170, 520, 700, 30);
opt1.setBackground(Color.WHITE);
opt1.setFont(new Font("Dialog", Font.PLAIN, 20));
add(opt1);

opt2 = new JRadioButton();
opt2.setBounds(170, 560, 700, 30);
opt2.setBackground(Color.WHITE);
opt2.setFont(new Font("Dialog", Font.PLAIN, 20));
add(opt2);

opt3 = new JRadioButton();
opt3.setBounds(170, 600, 700, 30);
opt3.setBackground(Color.WHITE);
opt3.setFont(new Font("Dialog", Font.PLAIN, 20));
add(opt3);

opt4 = new JRadioButton();
opt4.setBounds(170, 640, 700, 30);
opt4.setBackground(Color.WHITE);
opt4.setFont(new Font("Dialog", Font.PLAIN, 20));
add(opt4);

groupoptions = new ButtonGroup();
groupoptions.add(opt1);
groupoptions.add(opt2);
groupoptions.add(opt3);
groupoptions.add(opt4);

next = new JButton("Next");
next.setBounds(1100, 550, 200, 40);
next.setFont(new Font("Tahoma", Font.PLAIN, 22));
next.setBackground(new Color(30, 144, 255));
next.setForeground(Color.WHITE);
next.addActionListener(this);
add(next);

lifeline = new JButton("50-50 Lifeline");
lifeline.setBounds(1100, 630, 200, 40);
lifeline.setFont(new Font("Tahoma", Font.PLAIN, 22));
lifeline.setBackground(new Color(30, 144, 255));
lifeline.setForeground(Color.WHITE);
lifeline.addActionListener(this);

```

B. SRCEENSHOTS:

