

MINI PROJECT-II

(2022-23)

‘Quizzzry’

Project Report



Institute of Engineering & Technology

Submitted By-

Ajay Singh (201500050)

Under the Mentorship Of-

Mr. Ankit Arora

(Technical Trainer)

Department of Computer Engineering & Applications

Declaration

I hereby declare that the work which is being presented in the Bachelor of technology. Project 'Quizzzry', in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the Mentorship of Mr Ankit Arora, Mentor, Dept. of CEA, GLA University.

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign:

Name of Candidates: Ajay Singh (201500050)

Certificate

This is to certify that the project entitled 'Quizzzry', carried out in Mini Project - II Lab, is a Bonafede work by Ajay Singh is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Mentor:

Name of Mentor: Mr. Ankit Arora

Date :

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the mentor Mr. Ankit Arora, our Mentor. He has been helping me since Day 1 in this project. He provided me with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meetings to check the progress of the project and providing us with the resources related to the project. Without his help, I wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign:

Name of Candidates:

Ajay Singh (201500050)

ABSTRACT

‘Quizzzry’ web app is designed to provide an engaging and educational experience for users. It allows users to test their knowledge on a particular topic through a range of well-crafted questions of varying difficulty levels. The app provides instant feedback and tracks the user's progress, motivating them to improve their score. The web app is built on Node.js, allowing for a fast and scalable back-end, and uses Firebase Authentication for secure user authentication. The quiz web app is designed to be visually appealing and easy to use, providing a fun and interactive learning experience for users. Overall, the quiz web app aims to provide an enjoyable and educational experience for users through a well-crafted and user-friendly platform.

The app uses Firebase Authentication to ensure secure user authentication, protecting user data and information. Firebase Authentication allows users to sign up and log in using third-party authentication providers such as Google.

Overall, Quizzzry web app aims to provide an enjoyable and educational experience for users through a well-crafted and user-friendly platform. It is ideal for anyone looking to test their knowledge and learn something new on a particular topic..

INDEX

CONTENTS	PAGE NUMBERS	SIGNATURES
Cover page	1	
Declaration	2	
Certificate	3	
Acknowledgment	4	
Abstract	5	
Index	6	
Chapter 1:Introduction	7	
Chapter 2:Software Requirement Analysis	8-13	
Chapter 3:Software Design	13-14	
Chapter 4:Language, Technology and Tools used	14-17	
Chapter 5:Implementation and	17-18	

Interface		
Chapter 6:Testing	18-22	
Chapter 7: Conclusion	22	
References	23	

CHAPTER-1

INTRODUCTION

1.1 CONTEXT

In today's fast-paced world, where knowledge and information are readily available at our fingertips, quizzes have become an increasingly popular way to test one's knowledge and engage with a particular topic. Whether it's for education, entertainment, or just for fun, quizzes offer a simple and effective way to assess one's knowledge.

However, traditional quizzes can be limiting in terms of accessibility and engagement. Printed quizzes can only be distributed to a limited audience, and marking them can be time-consuming and cumbersome. Meanwhile, online quizzes often lack the interactivity and engagement necessary to keep users interested.

Built on Node.js and using Firebase Authentication, the app will provide secure and reliable user authentication, ensuring that user data and information remain protected. The app will also feature real-time progress tracking, allowing users to see how they're performing and where they need to improve.

Quizzzry aims to be an ideal tool for educators, trainers, and anyone looking to learn something new in an interactive and engaging way. Whether you're a student looking to test your knowledge or a teacher looking to supplement your lessons, the app will provide a fun and effective way to engage with a particular

topic.

1.2 MOTIVATION

The motivation behind creating Quizzzry from a desire to provide an engaging and interactive way for people to test their knowledge on a variety of topics. Quizzes have always been a popular way to engage with information and learn something new, but traditional methods of distributing and grading quizzes can be cumbersome and limiting.

1.3 OBJECTIVES

- Provide anime fans with a comprehensive collection of Quizzes.
- Create a user-friendly interface that allows users to Login, create quizzes, join quizzes etc.
- Optimize the platform's performance by utilizing modern web technologies such as JavaScript, Firebase, NodeJs etc.

1.4 SOURCE

The source of our project (including all the project work, documentations and

presentations) will is available at the following link :

<https://github.com/Ajayvirmoti/quizzzery>

CHAPTER -2

SOFTWARE REQUIREMENT ANALYSIS

2.1 IMPORTANCE OF Quizzzry

Quizzzry quiz app is important because it provides a platform for users to test their knowledge and learn something new in an engaging and interactive way. Quizzes have always been a popular way to assess knowledge and are an effective tool for learning, but traditional methods of distributing and grading quizzes can be limiting.

With the increasing popularity of online learning and remote education, your quiz app offers a convenient and accessible solution to the limitations of traditional quiz methods. The app allows users to access a wide range of well-crafted questions on a variety of topics, catering to users of all knowledge levels.

2.2 PROBLEM STATEMENT

Traditional methods of distributing and grading quizzes can be cumbersome and limiting, especially with the increasing popularity of online learning and remote education. While quizzes have always been a popular way to assess knowledge and are an effective tool for learning, they often lack the interactive and engaging features that can make learning more fun and effective.

Furthermore, existing quiz apps often lack the customization and personalization features that cater to different audiences and knowledge levels. This can lead to users becoming disengaged and losing interest in the quiz experience.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement

Processor : Intel or AMD (1.3 GHz or above)

Operating System : Any Operating System

RAM: 2 GB (or higher)

Hard disk: Not Explicitly Required

Software Requirement

Software used (to Build): NeoVim and VS Code

Software used (to Run): Any Browser

Language	used:	JavaScript,	HTML
-----------------	--------------	-------------	------

Database: Firebase

2.4 FUNCTIONALITIES

1. User Authentication: The app should provide user authentication

functionality to ensure that only authorized users can access the quizzes.

Users can create accounts or log in using their social media accounts to access the app.

1. **Quiz Creation:** The app should allow the creation of quizzes by users with administrative access. This functionality should include adding questions, answers, and explanations, and assigning the quiz to specific categories or topics.
2. **Quiz Taking:** The app should allow users to take quizzes assigned to them, displaying one question at a time with multiple-choice answers. Users should be able to select their answers and move on to the next question.
3. **Real-time Progress Tracking:** The app should provide real-time progress tracking for users taking quizzes. Users should be able to see how many questions they have answered, how many questions are left, and their current score.
4. **Leaderboard:** The app should display a leaderboard showing the top scorers for each quiz or category. Users should be able to see where they rank compared to other users and be motivated to improve their scores.
5. **Categories and Topics:** The app should provide functionality to organize quizzes into categories or topics. Users should be able to browse and select quizzes based on their interests or areas of expertise.
6. **Customization:** The app should provide customization functionality to cater to different audiences and knowledge levels. This functionality

should include selecting the number of questions, the difficulty level, and the type of quiz (timed or untimed).

7. Feedback and Explanation: The app should provide feedback and explanation functionality after each question is answered. Users should be able to see the correct answer and an explanation of why it is correct.
8. Results and Analysis: The app should provide functionality to display the results of the quiz, including the user's score, the number of correct and incorrect answers, and a detailed analysis of the user's performance.
9. Sharing: The app should allow users to share their quiz results or progress on social media platforms or with their friends and family.

WEBSITE

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have 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

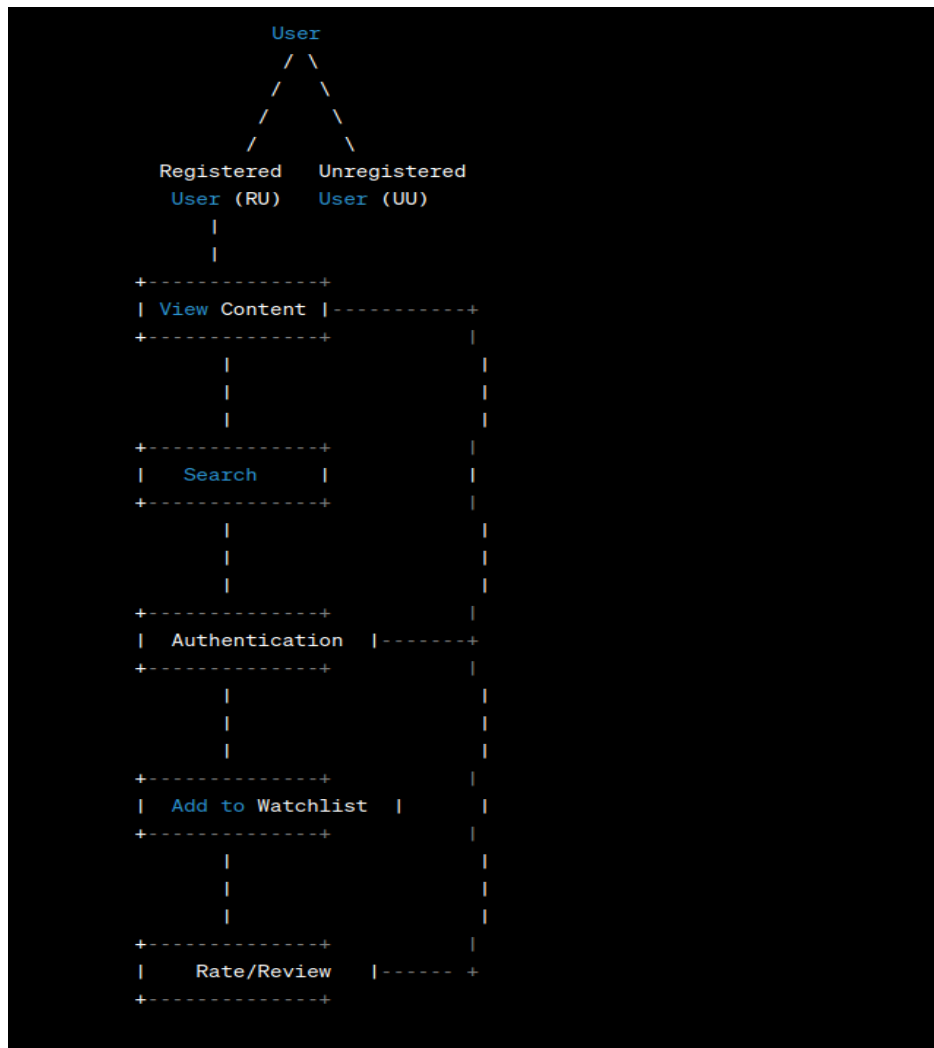
following points-

- 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, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- 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

CHAPTER-3

SOFTWARE DESIGN

3.1 USE-CASE DIAGRAM:



CHAPTER-4

LANGUAGES, TECHNOLOGY AND TOOLS USED

4.1 HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by

Hypertext Markup Language, and Web page.

Hyper Text: HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Web Page: A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages.**

4.2 CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user

interfaces for web applications and user interfaces for many mobile applications.

4.3 JavaScript

JavaScript is a high-level programming language used to create dynamic, interactive websites and web applications. It is widely used in front-end web development to add interactive elements to web pages, such as forms, animations, and navigation menus. JavaScript is also used in back-end web development with frameworks like Node.js to build scalable and efficient server-side applications. It is a powerful language that supports object-oriented, functional, and procedural programming paradigms. With its versatility, popularity, and large community of developers, JavaScript has become a crucial skill for anyone interested in web development.

4.5 API

API is the abbreviation of the term *Application Programming Interface*. It is the software responsible for the connection for the communication and information exchange between two apps. API connects two devices or programs in order to facilitate the exchange of information between them. It is the interface that serves the other parts of the software. The API specifications are the standards or documents designed to describe the creation of such connections. If a computer system meets these standards, then it is said to expose an API. The specification or implementation both are known as the API.

4.6 Tailwind

Tailwind CSS is a utility-first CSS framework designed to enable users to create applications faster and easier. You can use utility classes to control the layout, color, spacing, typography, shadows, and more to create a completely custom component design — without leaving your HTML or writing a single line of custom CSS.

4.7 GITHUB

GitHub is an immense platform for code hosting. It supports version controlling and collaboration and allows developers to work together on projects. It offers both distributed version control and source code management (SCM) functionality of Git. It also facilitates collaboration features such as bug tracking, feature requests, task management for every project.

Essential components of the GitHub are:

- Repositories
- Branches
- Commits
- Pull Requests
- Git (the version control tool GitHub is built on)

4.9 Firebase:

Firebase is a modern database toolkit that provides a type-safe and intuitive way to access databases from application code. It allows developers to easily work with databases and perform common database operations using a clean, type-safe, and expressive API.

CHAPTER-5

IMPLEMENTATION AND INTERFACE

5.1 IMPLEMENTATION

Javascript is a scripting language used to enhance the functionality of the browser. Java script is integrated with HTML and navigator 2.02. JavaScript facilitates the developer with properties related to document windows, frames, loaded documents, and links.

5.2 USER INTERFACE DESIGN

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventual presentation of desired inputs and outputs. The

overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly.
5. Default values for fields and answers to be entered by the user should be specified.
6. A user should not be allowed to proceed without correcting an error.
7. The system user should never get an operating system message or fatal error.

CHAPTER-6

TESTING

Implementation and Software Specification Testing

Detailed Design of Implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

Installation Phase

In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

System Installation

The process of starting the actual use of a system and training user personnel in its operation.

Review Phase

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized Transystem in terms of benefits and savings projected at the start of the project.

Development Recap

A review of a project immediately after completion to find successes and potential problems in future work.

Post-Implementation Review

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

THE STEPS IN THE SOFTWARE TESTING

The steps involved during Unit testing are as follows:

- a. Preparation of the test cases.
- b. Preparation of the possible test data with all the validation checks.
- c. Complete code review of the module.
- d. Actual testing done manually.

e. Modifications done for the errors found during testing.

f. Prepared the test result scripts.

The unit testing done included the testing of the following items:

1. Functionality of the entire module/forms.
2. Validations for user input.
3. Checking of the Coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System

Integration, We integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

The steps involved during System testing are as follows:

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.

- . Actual testing done manually.
- Recording of all the reproduced errors. Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.

The System Testing done included the testing of the following items:

1. Functionality of the entire system as a whole.
2. User Interface of the system.
3. Testing the dependent modules together with all the possible test data scripts.
4. Verification and Validation testing.
5. Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with application. Thus, we reached the final phase of the project.

There are other six tests, which fall under special category. They are described below:

- Peak Load Test: it determines whether the system will handle the

volume of handles of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.

- Storage Testing: It determines the capacity of the system to store transaction data on a disk
 - or in other files.
 - Performance Time Testing: it determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.
 - Recovery Testing: This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.
 - Procedure Testing: It determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down
 - system at the end of week or responding to paper-out light on printer.
- Human Factors Testing: It determines how users will use the system when processing data or preparing reports.

CHAPTER-7

CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have 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 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, Scope, and applicability.

- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that
 - can be done on these things.
- 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.

REFERENCES

- Project Link: <https://github.com/Ajayvirmoti/quizzzery>
- Next JS Docs : <https://nextjs.org/docs/getting-started>

- Tailwind css : <https://tailwindcss.com/docs/configuration>
- and some blogs related to the topics or concepts.