

A PROJECT REPORT ON

Quiz Application

Submitted By:

Govind Sharma - G - 2115000419

Kanishk Patel - D - 2115000505

Ritik Chauhan -N - 2115000853

***In partial fulfilment for the award of the degree of***

# Bachelor of Engineering

## IN

**Computer Science**

# BONAFIDE CERTIFICATE

Certified that this project report **“Quiz Application”** is the bonafide work of **“Govind Sharma, Kanishk Patel , Palak Dhakrey** and **Ritik Chauhan”** who carried out the project work under my/our supervision.

|  |  |
| --- | --- |
| **SIGNATURE**  **HEAD OF THE DEPARTMENT**  **Mr. Rohit Agrawal** | **SIGNATURE**  **SUPERVISOR**  Technical Trainer:  **Mr. Mohd. Aslam**  Training and development |

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

The project work in this report is an outcome of continuous work over a period and drew intellectual support from various sources. We would like to articulate our profound gratitude and to all those people who extended their wholehearted co- operation and have helped us in completing this project successfully.

We are thankful to our mentor Mr. Mohd Aslam for teaching and assisting us with the new technology and guiding us at every step, and it wouldn’t have been possible for us to finish the project in such short period of time if it were not for his motivation. I would also like to thank all the faculty members who were directly and indirectly contributed in the completion of our project .

# TABLE OF CONTENTS

**List of Figures** Error! Bookmark not defined.

**Abstract** Error! Bookmark not defined.

## Chapter 1 Introduction

* 1. **Client Identification**

## Identification of Problem

* 1. **Identification of Task**

## Timeline

* 1. **Organization of the Report**

## Chapter 2 Literature Survey

**2.1 Analysis**

## Chapter 3 Design Flow / Processes

* 1. **Concept Generation**

## Evaluation & Selection of specifications / Features

* 1. **Analysis of Specifications and features**

## Design Flow

* 1. **Implementation Plan**

## Front-end tech

* + 1. **Back-end tech**

## Chapter 4 Result Analysis and Validation

* 1. **Sign up page**

## Admin page

* 1. **User page**

## Reports page

## Chapter 5………………Performance Analysis

## 5.1 Agile Methodology

## 5.2 Performance Analysis

## Chapter 5 Conclusion & Future Work

* 1. **Conclusion**

## Future Work

# ABSTRACT

In the realm of modern education, technological advancements have significantly reshaped traditional teaching methodologies. This mini project focuses on the design and implementation of an Online Quiz Platform, aimed at providing an engaging and interactive learning experience for users. The application is designed to cater to educational institutions, training programs, and online courses, offering a flexible platform for creating and administering quizzes.

The Online Quiz Platform employs a user-friendly interface, allowing both instructors and students to seamlessly navigate through various features. Instructors can create diverse sets of quizzes with different question types, including multiple-choice, true/false, and open-ended questions. The system supports the incorporation of multimedia elements to enhance the overall learning experience.

# CHAPTER 7

**INTRODUCTION**

## Client Identification/Need Identification/Identification of relevant

## In this phase, we aim to identify key stakeholders for our Online Quiz Platform, understanding their roles and expectations. Simultaneously, we'll conduct a needs assessment, pinpointing user requirements and preferences. Relevant technologies and emerging trends in online learning will be identified, ensuring our quiz platform aligns with the current educational landscape. This process will serve as the foundation for developing a user-centric, technologically advanced, and educationally effective online quiz platform.

## Identification of Problem

## Our Online Quiz Platform addresses several challenges in traditional assessment methods, including limited interactivity, time-consuming grading processes, and restricted accessibility. It resolves issues related to engaging learners effectively in remote or online environments, providing real-time feedback, and accommodating various learning styles. Additionally, the platform aims to overcome barriers to traditional assessment by offering a dynamic and user-friendly experience that enhances the overall educational process.

## Identification of Task

The main objective of this project is to Build an online quiz platform that allows users to create and take quizzes on various topics, with scoring and generate reports of their scores.

## Timeline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task / Period | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
| Project Selection |  |  |  |  |  |
| Mentor Allocation |  |  |  |  |  |
| Project Planning |  |  |  |  |  |
| Prototype and Designing |  |  |  |  |  |
| Documentation |  |  |  |  |  |

* 1. **Organization of the Report**

GLA University

# CHAPTER 2

**LITERATURE SURVEY**

Numerous studies have demonstrated the positive impact of note-sharing, resource access, and collaborative learning on student learning and success. These strategies provide students with opportunities to learn from each other, access a wealth of educational resources, and enhance their note-taking skills. Academies, by incorporating these elements, has the potential to revolutionize the college learning experience.

Significance of Online Quiz Platform in the Modern Online Learning:

1. **Positive impact on student learning**: Studies have shown that Online Quiz

platform saves the time of both teachers and students for preparation. So in less time

they get

1. **Evolution of Assessment Methods:** The literature reveals a shift from traditional assessment methods to online alternatives. Studies by Smith et al. (2018) and Johnson (2020) highlight the growing need for dynamic and interactive assessment tools to accommodate diverse learning styles.
2. **User-Centric Design:** Research by Chen et al. (2019) emphasizes the importance of user-centric design in online educational platforms. Understanding user preferences and incorporating features that enhance the user experience are crucial considerations, aligning with the goals of our Online Quiz Platform.
3. **Technological Integration:** The work of Wang and Zhang (2021) underscores the significance of integrating emerging technologies, such as artificial intelligence and machine learning, into quiz platforms. These technologies can optimize question generation, grading processes, and overall platform efficiency.
4. **Assessment in Remote Learning Environments:** The COVID-19 pandemic has accelerated the adoption of online education. Studies by Garcia et al. (2020) and Khan (2021) discuss the challenges and opportunities in assessing students remotely, emphasizing the need for secure and reliable online quiz platforms.
5. **Real-Time Feedback:** Providing instant feedback is a key aspect of effective online assessment. Insights from the work of Anderson and Mitchell (2017) and Li et al. (2019) highlight the positive impact

# CHAPTER 3

**DESIGN FLOW/PROCESS**

## Concept Generation

## The concept behind our Online Quiz Platform is rooted in the vision of creating a dynamic and engaging space for educational assessment. Focused on transforming traditional quiz-taking into a modern, interactive experience, the platform emphasizes the following key concepts

## Engaging User Interface

## Real-Time Feedback

## Secure and Reliable Assessment

## Evaluation & Selection of Specifications/Features

## In the process of evaluating and selecting specifications/features for the Online Quiz Platform, a meticulous assessment was conducted based on key criteria to ensure the platform's effectiveness and user satisfaction.

## Analysis of Specifications and Features

We have worked on modern frameworks to ensure the compatibility of our website across various devices. We have also implemented encryption to data protection regulations for secure Authentication integration. We have added JWT authentication for secure password of users

Other than that we have used optimized search algorithms for quick filtering of study resources according to the year wise student and providing relevant results with less delay.

## Design Flow

1. **Registration and login page**: It has a consistent layout for registration and login for both user and admin
2. **Home Page**: It is a dynamic layout that is different for user and admin in user panel you will not get options like add quizzes add question and view reports of all users
3. **Exams**: this is only for admin to add quizzes and questions to that quiz
4. **Reports** It provides the reports of user and admin can view the reports of all user but user can view only his/her report

## Implementation Plan

We have implemented the designing of our website with the latest tools available to maintain the compatibility of our website with the modern tech.

Below is the description of the tech stack that we have used in our project.

Technologies used:

Our project is based on front-end and back-end technologies.

**3.5.1.** For front-end we are using React JS.

**React JS** is a JavaScript library that is widely used for building user interfaces is one of the most popular front-end development frameworks, and it provides a rich set of tools and features that allow developers to customize the behavior and appearance of their user interfaces. With React, developers can create unique and engaging user experiences that are tailored to the needs of their users.

React is useful in a project because it allows developers to build complex user interfaces with ease. By using React, developers can create reusable UI components that can be combined to create a variety of different layouts and user experiences. This makes it easier to build *scalable* and *maintainable web applications*.

One of the key features of React is its ability to provide real-time updates to the user interface. By using a technique called "virtual DOM," React can update the user interface without having to reload the entire page. This helps to create a seamless user experience that is *faster* and *more responsive* than traditional web applications.

Another benefit of using React in a project is its ability to work with other JavaScript libraries and frameworks. React is designed to be modular and can be easily integrated with other libraries and frameworks, such as Redux, React Router, and Axios. This allows developers to build powerful and complex web applications that can meet the needs of their users.

React is also useful in a project because it is highly customizable. React provides a rich set of tools and features that allow developers to customize the behaviour and appearance of their user interfaces. With React, developers can create unique and engaging user experiences that are tailored to the needs of their users.

In conclusion, React is a powerful and flexible JavaScript library that is useful in a wide range of web development projects. Its ability to provide real-time updates, work with other libraries and frameworks, and provide a high degree of customization makes it an essential tool for building scalable and maintainable web applications. Whether you're building a small website or a large web application, React is a valuable tool that can help you create a great user experience for your audience.

* + 1. For back-end, Node JS, Express JS, MongoDB and Mongoose are used.
       1. **Node JS** is an open-source, cross-platform, back-end JavaScript runtime environment that allows developers to build fast and scalable applications. Node JS interacts with the server to handle the client requests while utilizing the commands that are being designed in Express JS.

It uses an event-driven, non-blocking I/O model that makes it highly efficient for building real-time applications, especially those that involve large amounts of data.

It is useful in a project in several ways, here are some of them:

* + - * 1. **High performance**: It is built on the V8 JavaScript engine, which is the same engine used by Google Chrome. This makes Node.js *highly efficient* and allows it to handle a large number of requests without affecting the performance of the application.
        2. **Scalability**: It is highly scalable, which means *it can handle a large number of connections simultaneously without slowing down*. This makes it ideal for building applications that need to handle a large amount of traffic.
        3. **Single language**: With Node.js, developers can use JavaScript on both the front-end and back-end, making it easy to develop full-stack applications using a single language.
        4. **Large community**: Node.js has a large and active community of developers who contribute to the development of various libraries and modules. This makes it easy for developers to find solutions to their problems and improve the quality of their applications.

Node.js is a powerful tool that can help developers build fast, scalable, and efficient applications. Its ease of use and large community make it an attractive choice for developers who want to build full-stack applications using a single language.

* + - 1. **Express JS** is a popular open-source web application framework built on top of Node.js. It provides a set of tools and utilities for building web applications and APIs in Node JS. Express JS is known for its simplicity, flexibility, and scalability, and it is widely used by developers to build web applications, APIs, and micro services.

It provides light-weight framework of Node JS and with it we can perform crud operations and create our own server easily fulfilling our project requirements. It also acts as a *powerful middleware for a range of operations such as authentication, logging and error handling*.

Some of the key advantages of using Express JS includes:

* + - * 1. **Simplicity**: Express JS is a lightweight framework that provides a simple and intuitive API for building web applications and APIs. It has a minimalist design philosophy that emphasizes simplicity and ease-of-use.
        2. **Flexibility**: It is highly flexible and customizable, allowing developers to build web applications and APIs that meet their specific needs. It provides a modular architecture that allows developers to use only the components they need and replace or extend them as needed.
        3. **Scalability**: It is highly scalable and can handle high volumes of traffic and requests. It provides a non-blocking I/O model that enables asynchronous processing and supports clustering for horizontal scaling.
        4. **Middleware**: It provides a powerful middleware system that allows developers to easily add features and functionality to their web applications and APIs. Middleware functions can be used for a wide range of purposes, such as authentication, logging, error handling, and more.
        5. **Routing**: It also provides a powerful routing system that allows developers to easily define the routes for their web applications and APIs. It supports a wide range of HTTP methods, such as GET, POST, PUT, DELETE, and more.
        6. **Integration**: Express.js integrates seamlessly with other popular Node.js libraries and tools, such as MongoDB, Socket.io, and more.

Express.js is a powerful and flexible web application framework that provides a wide range of benefits for developers. It enables developers to build fast, scalable, and customizable web applications and APIs with ease, making it an ideal choice for projects of all sizes and complexities.

* + - 1. **MongoDB** is a popular NoSQL document-oriented database that provides a flexible, scalable, and high-performance solution for handling unstructured or semi-structured data. It stores data in JSON-like documents with dynamic schemas, making it easy to store and retrieve complex data structures. Unlike traditional relational databases, MongoDB does not requires predefined tables to store data, which makes it ideal for handling large amount of data.

Some of the key features of MongoDB are:

* + - * 1. **Schema-less Database**: It is the great feature provided by the MongoDB. A Schema-less database means one collection can hold different types of documents in it. Or in other words, in the MongoDB database, a single collection *can hold multiple documents and these documents may consist of the different numbers of fields, content, and size*. It is not necessary that the one document is similar to another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases.
        2. **Document Oriented**: In MongoDB, all *the data stored in the documents* instead of tables like in RDBMS. In these documents, the data is stored in fields (key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.
        3. **Indexing**: In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.
        4. **Scalability**: MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.
        5. **Replication**: MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.
        6. **Aggregation**: It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods
        7. **High Performance**: The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.
      1. **Mongoose** is an Object Data Modeling (ODM) library for MongoDB. It defines a strongly-typed-schema, with default values and schema validations which are later mapped to a MongoDB document. It provides an incredible amount of functionality around creating and working with schemas. Mongoose currently contains eight Schema Types that a property is saved as when it is persisted to MongoDB. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

# CHAPTER 4

**RESULT ANALYSIS AND VALIDATION**

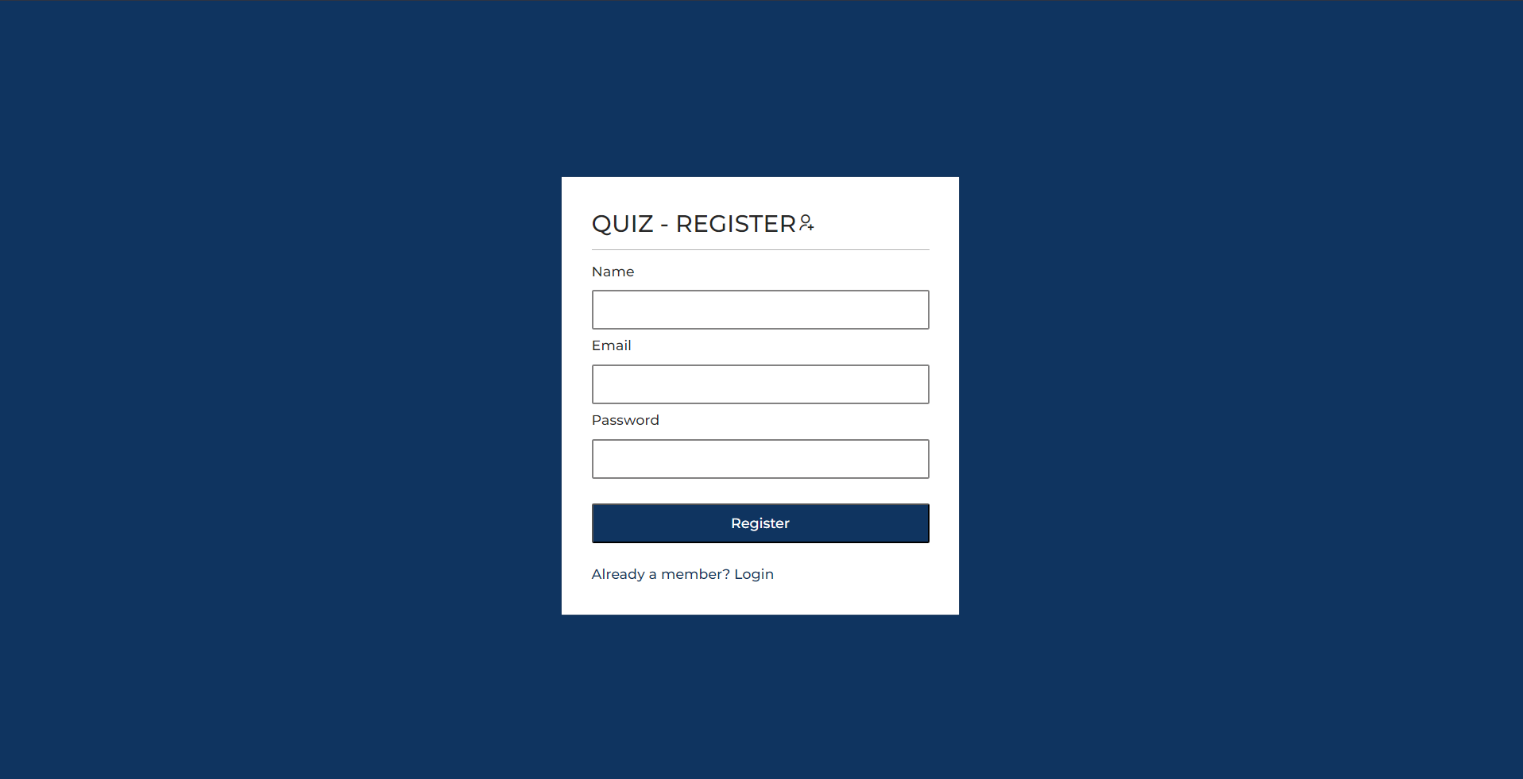


Figure 4.1 : Sign up

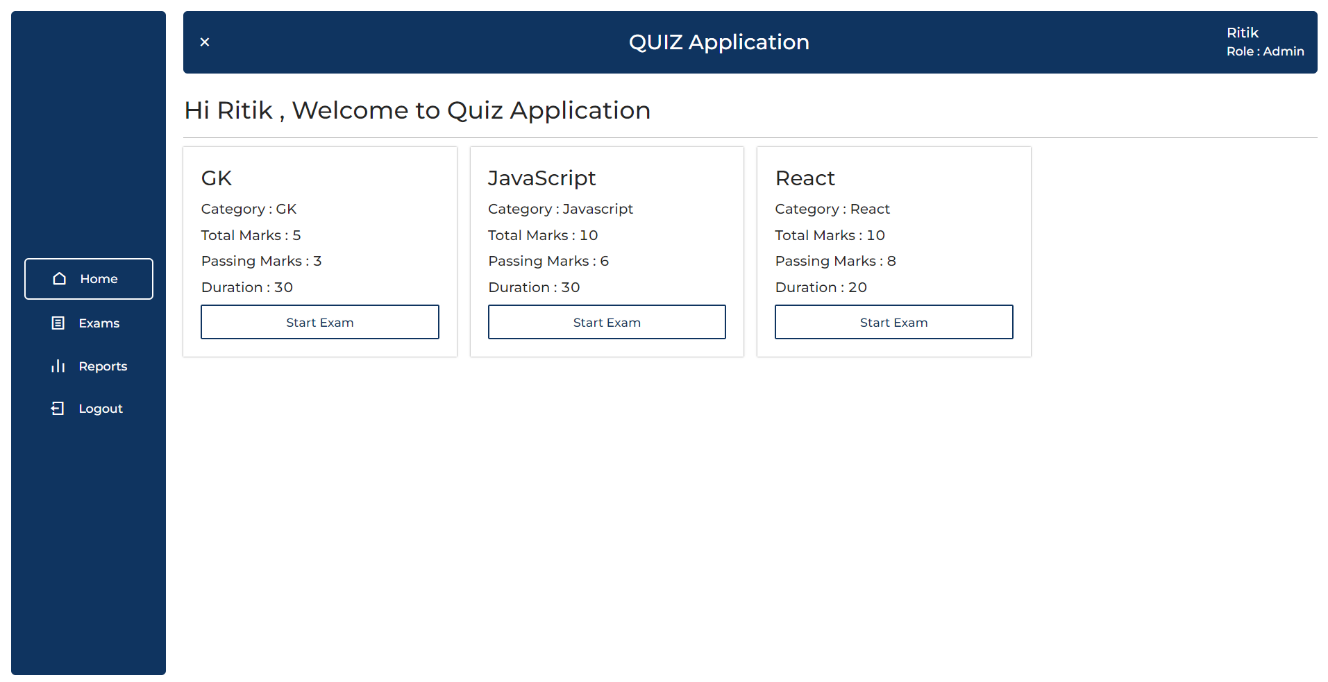


Figure 4.3: Admin Profile Home Page

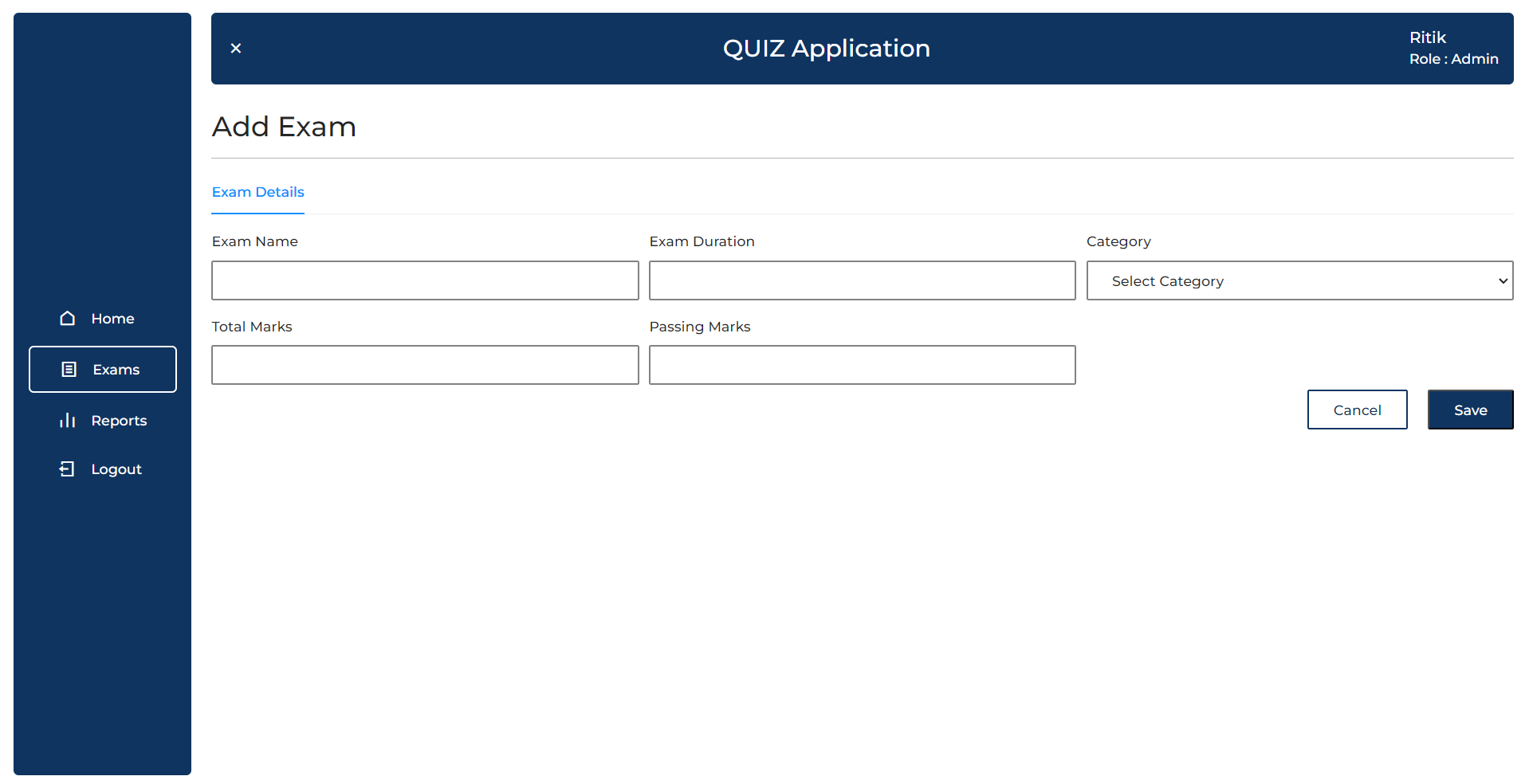


Figure 4.3: Add Quiz by Admin

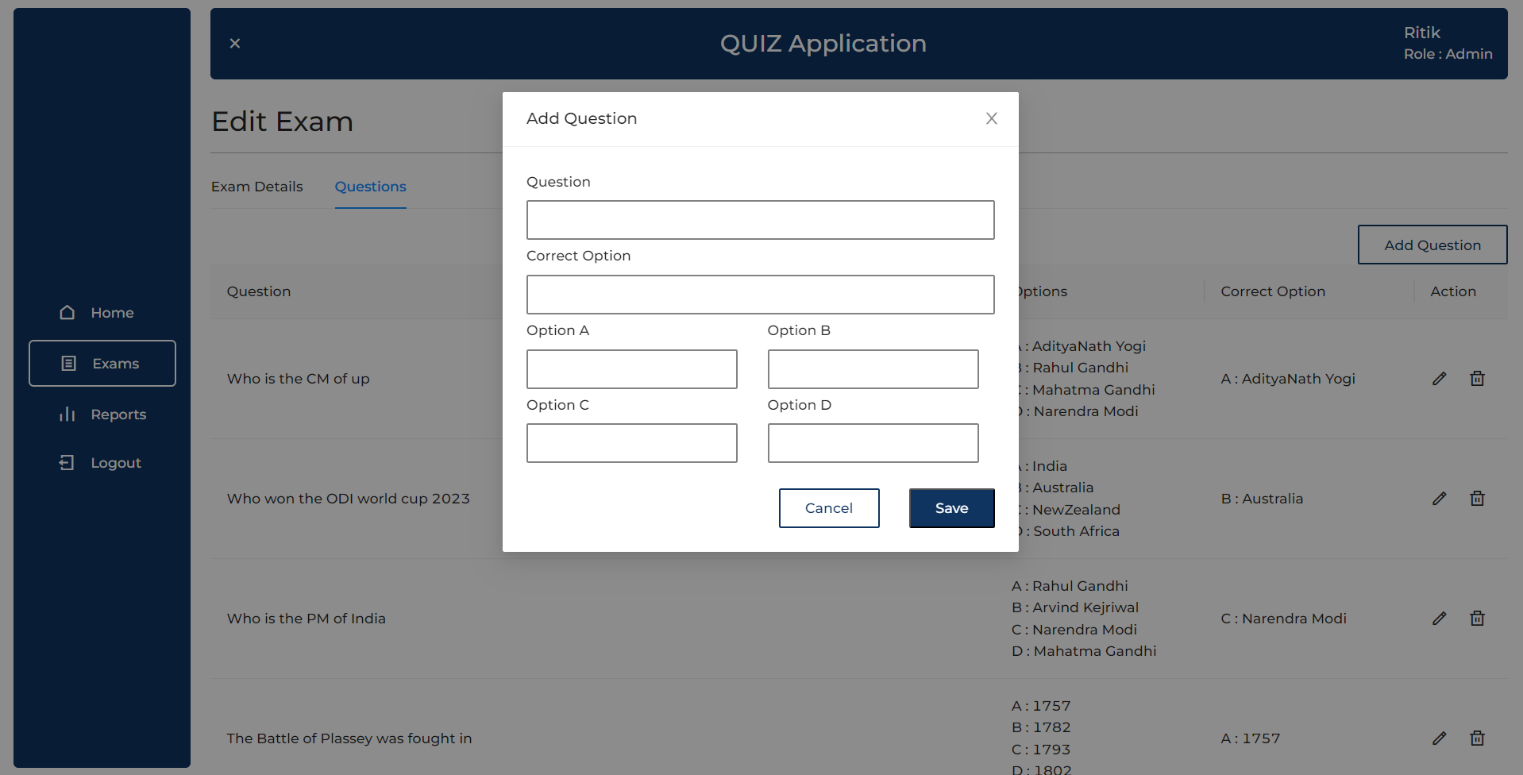


Figure 4.3: Add Questions in Quiz

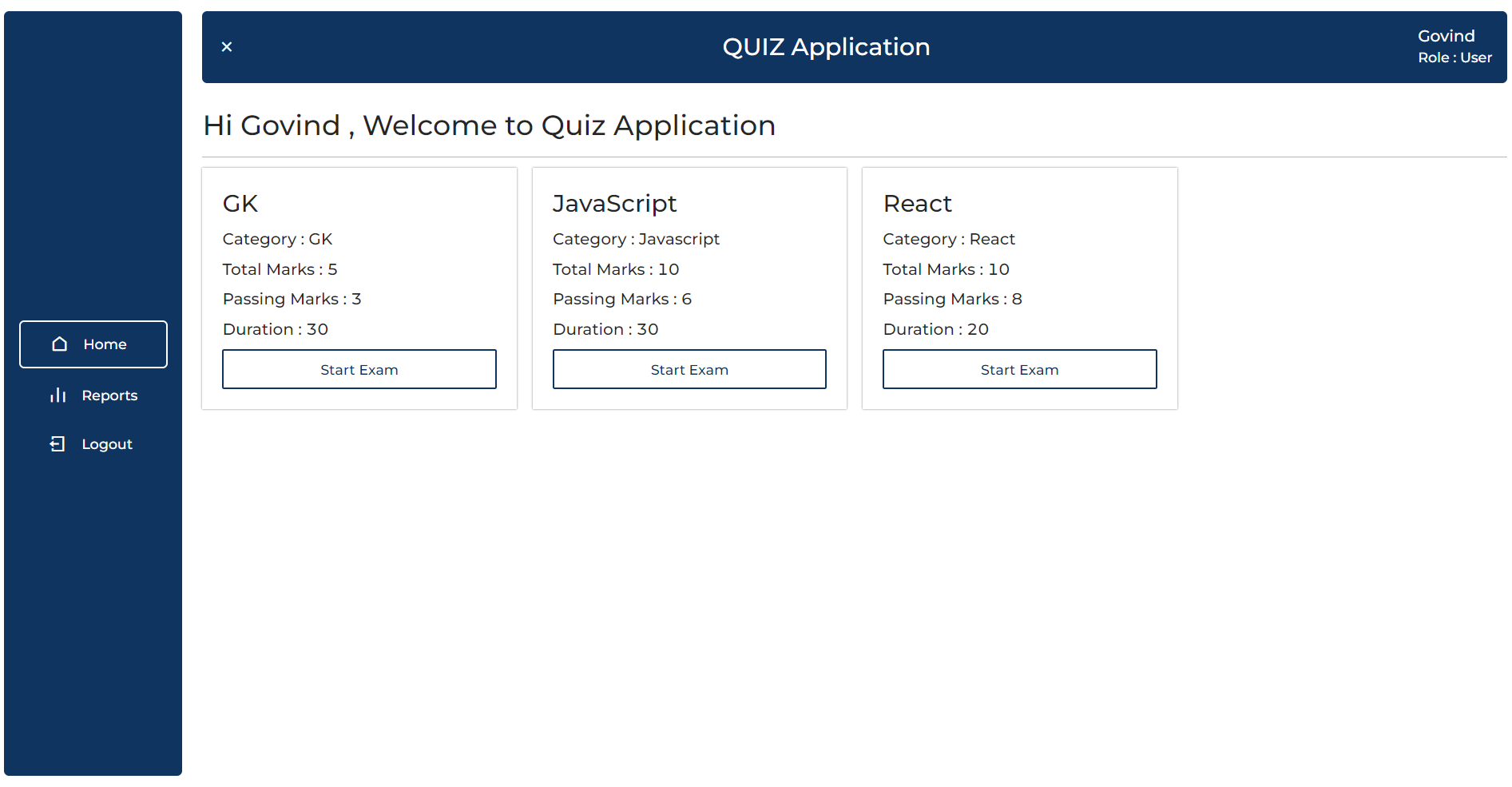


Figure 4.3: User’s Profile

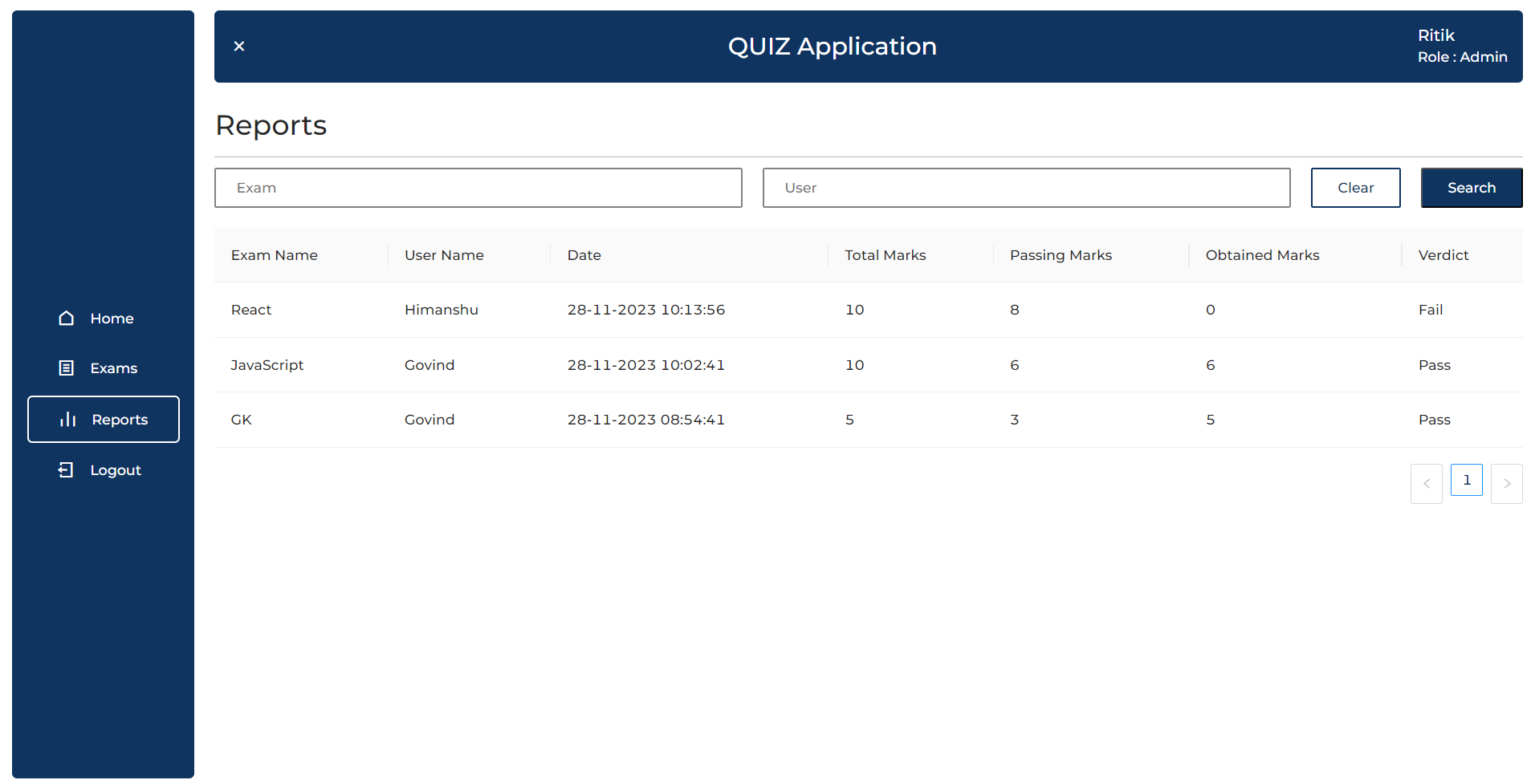


Figure 4.4: Reports

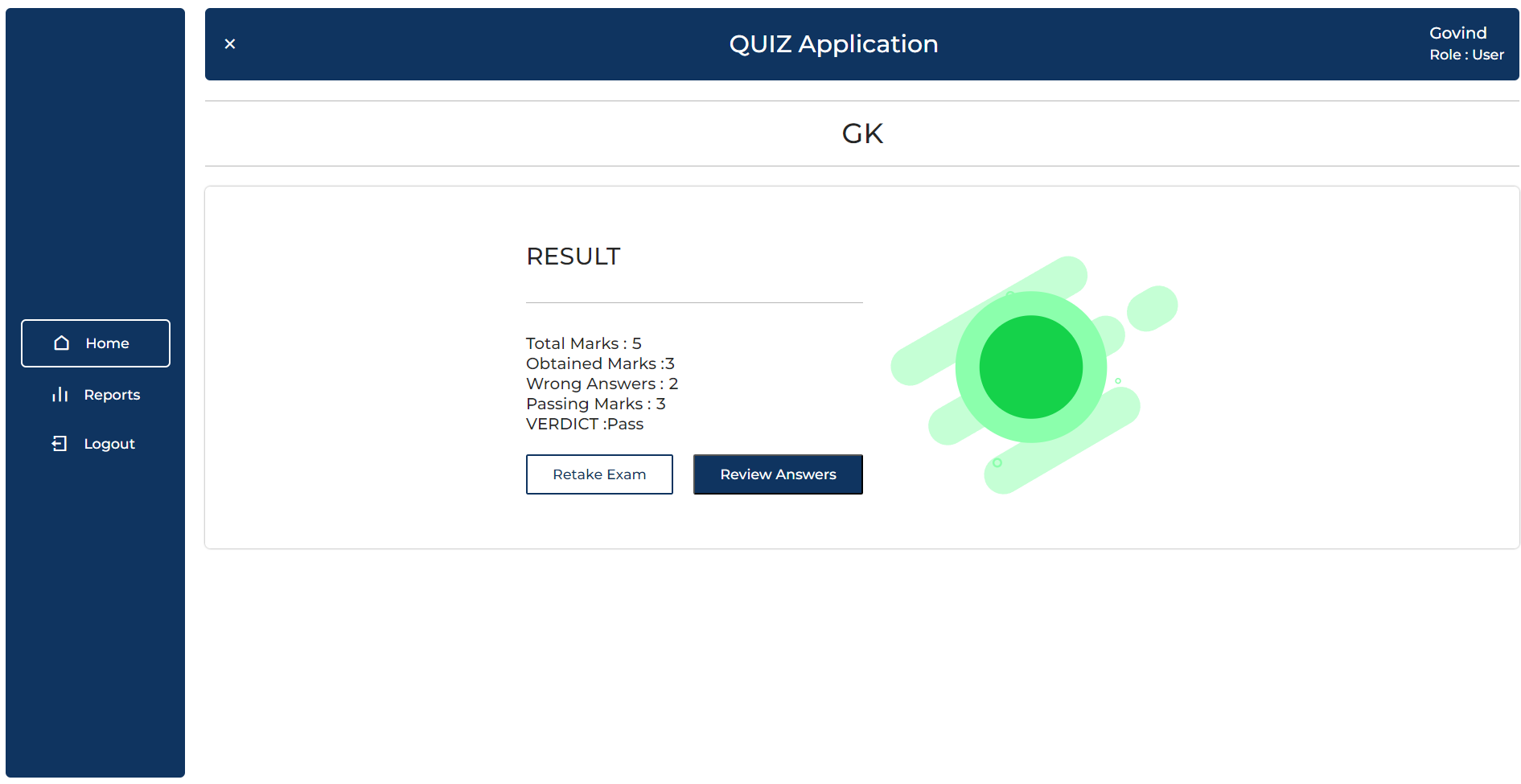


Figure 4.5 Result

# CHAPTER 5

**CONCLUSION AND FUTURE WORK**

## CONCLUSION

After completion of this project we have concluded that this web application works as

per the need and requirement of the client and is user friendly. Also this mini project helped me to understand the design, code and implementation processes which are performed while making any project. Many concepts were revised and many of them were very new which were learnt in making of this web application.

## FUTURE WORK

In future we will add some features like admin can create quiz by just giving the topic name

through AI so it will be time saving and we will also add feature to allow admins to add elements (images, videos) to questions, providing a richer quiz creation experience

# REFERENCES

* [**https://www.mongodb.com/docs/**](https://www.mongodb.com/docs/)
* [**https://devdocs.io/express/**](https://devdocs.io/express/)
* [**https://jwt.io/introduction**](https://jwt.io/introduction)
* [**https://docs.npmjs.com/**](https://docs.npmjs.com/)