

CSE 3102: Software Engineering Sessional

*AcademiaQuest*

Submitted By:

Md. Nihad Hossain (210201023)

**Year:** 3rd **Semester:** 1st

Department of Computer Science and Engineering (CSE)

Bangladesh Army University of Science and

Technology (BAUST), Saidpur

April 23, 2024

## Table of Contents

[Chapter 1: Introduction 3](#_Toc28393)

[Chapter 2: Project Feature Description 4](#_Toc28394)

[Chapter 3: Different Principles Used in the Project 11](#_Toc28395)

[Chapter 4: Project Time and Budget Management 13](#_Toc28396)

[Chapter 5: Conclusion 14](#_Toc28397)

# Chapter 1: Introduction

**AcademiaQuest** was conceived with the vision of revolutionizing learning by making it both enjoyable and accessible for students and educators. It aims to be a comprehensive web-based application tailored to the needs of universities, offering a plethora of educational resources and features. Developed using the ASP.NET framework and adhering to the Model-View-Controller (MVC) pattern, AcademiaQuest is poised to provide a seamless and user-friendly experience for all its users.

In today's fast-paced world, education stands as a cornerstone of societal progress and individual empowerment. However, traditional educational approaches often fall short in engaging students and maximizing learning outcomes. Recognizing this challenge, **AcademiaQuest** emerged as a pioneering initiative to redefine the educational landscape.

Driven by a passion for innovation and a commitment to excellence, **AcademiaQuest** aims to transcend the limitations of conventional learning methodologies. By harnessing the power of technology and leveraging the latest advancements in web development, we aspire to create a platform that not only facilitates learning but also inspires a lifelong love for knowledge.

At the heart of **AcademiaQuest** lies a simple yet profound philosophy: learning should be fun, accessible, and meaningful for all. Whether you're a student grappling with complex concepts, an educator seeking innovative teaching tools, or an administrator navigating the complexities of educational management, **AcademiaQuest** endeavors to be your trusted companion on the journey of learning and discovery.

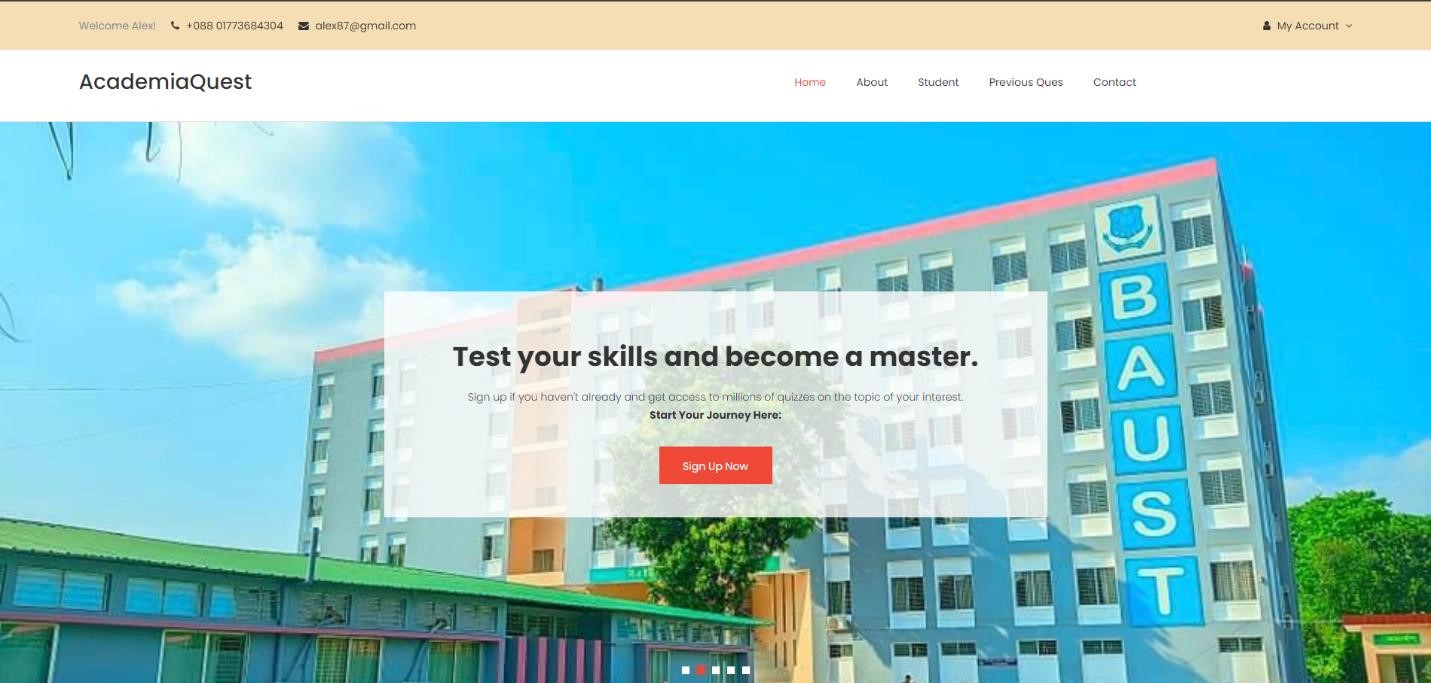


Fig.1 Homepage

# 

# Chapter 2: Project Feature Description

### 1. User Authentication

User authentication is a fundamental aspect of **AcademiaQuest,** ensuring that access to the platform is secure, controlled, and personalized for each user. In today's digital landscape, where privacy and security concerns are paramount, robust authentication mechanisms are essential to safeguarding sensitive user data and maintaining the integrity also intuitive, streamlined of the platform.

Fig.2

Register

or Login

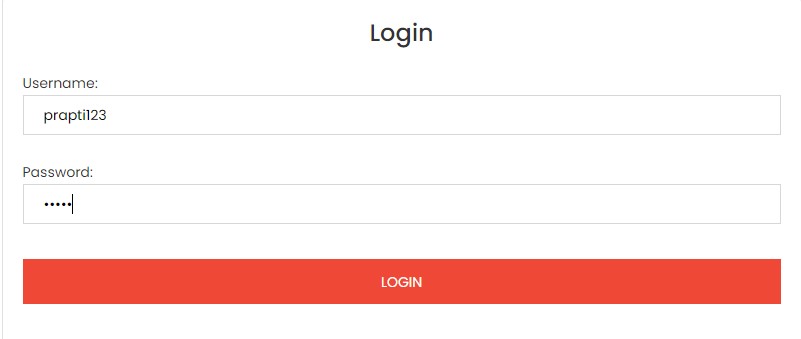
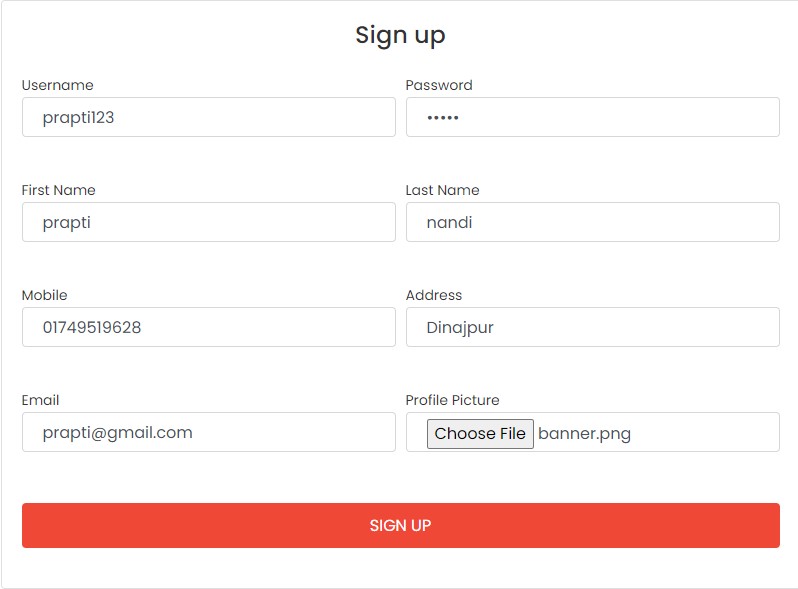
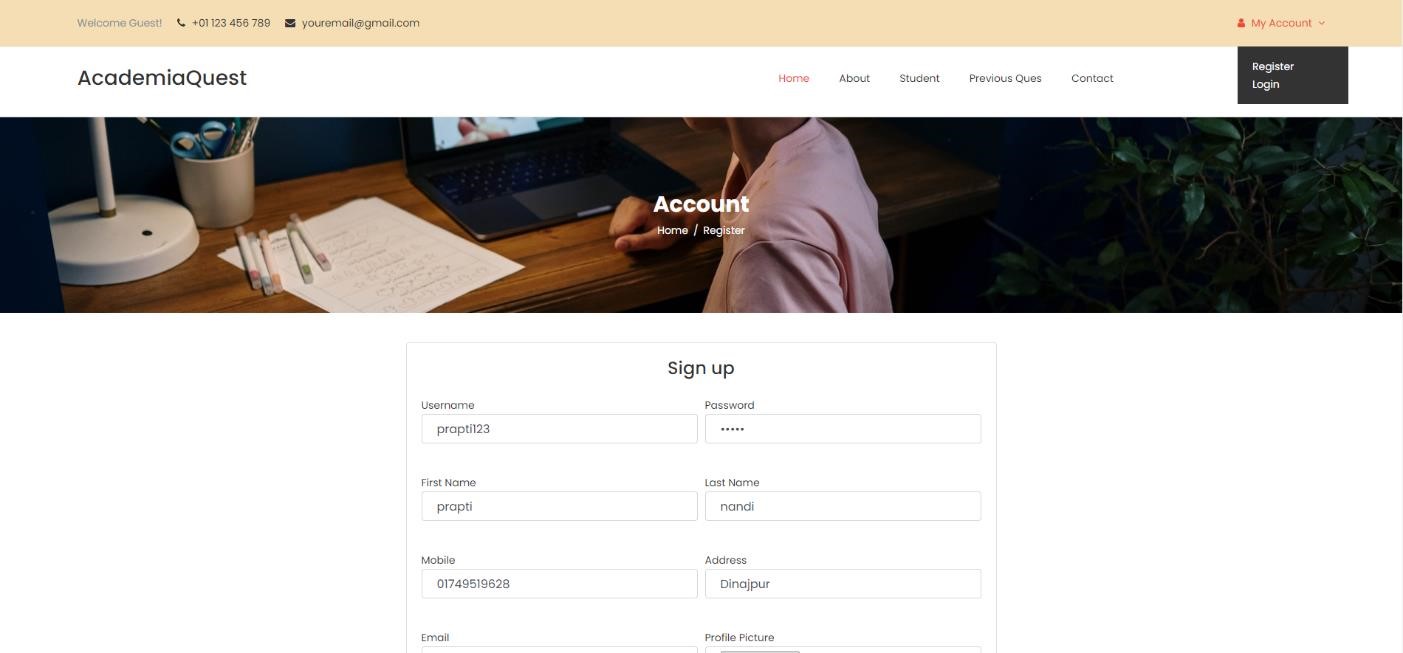


Fig.3 Register Fig.4 Login

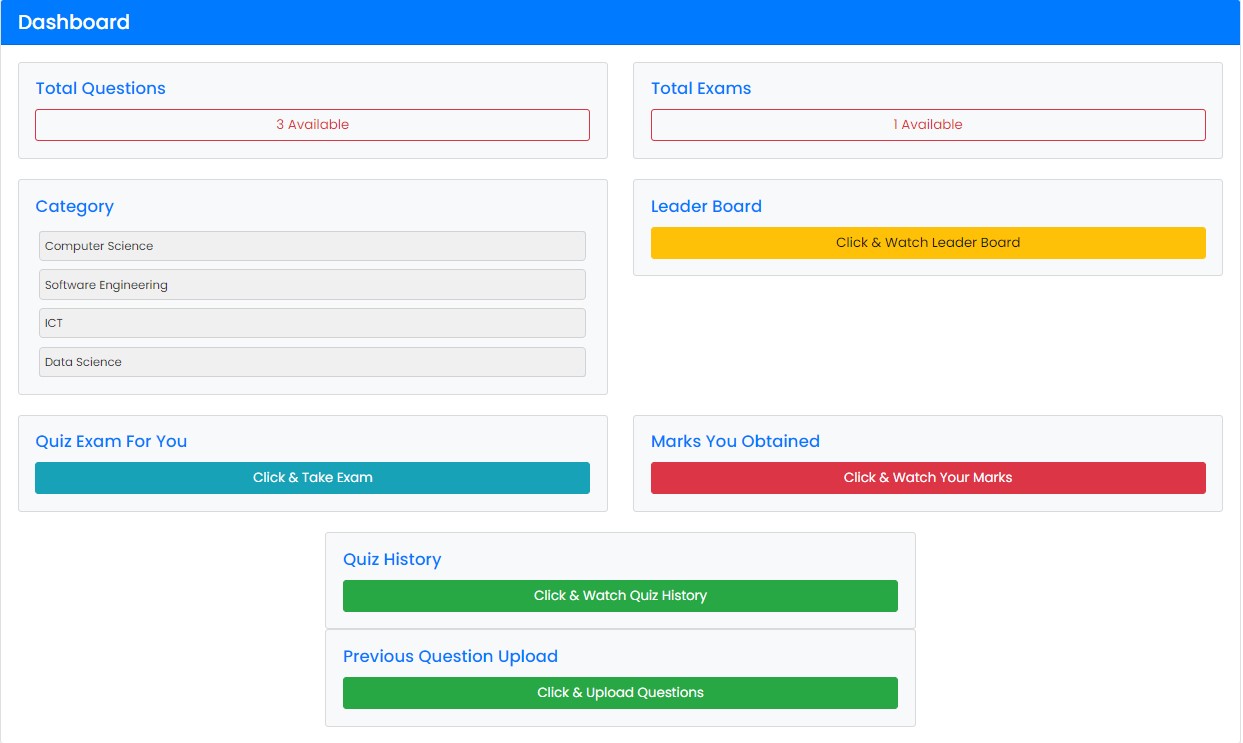
### 2.Student Dashboard

The student dashboard serves as the central hub of the **AcademiaQuest** platform, offering students a personalized and intuitive interface to manage their learning journey, track their progress, and access a wealth of educational resources. Designed with the needs and preferences of students in mind, the dashboard provides a seamless and immersive user experience, empowering students to take control of their learning and achieve their academic goals.

F

ig.5 Student d

ashboard



At the heart of the student dashboard lies a commitment to personalization and customization, ensuring that each student's learning experience is tailored to their unique needs, interests, and learning styles. Key features of the student dashboard include:

Total Questions: Here we can see all the questions that has been uploaded.

Category: It shows us different courses.

Quiz History: Here we can see our exam history.

Quiz Rating: We can also rate the quizzes.

Leader Board: Its for showing the ranking.

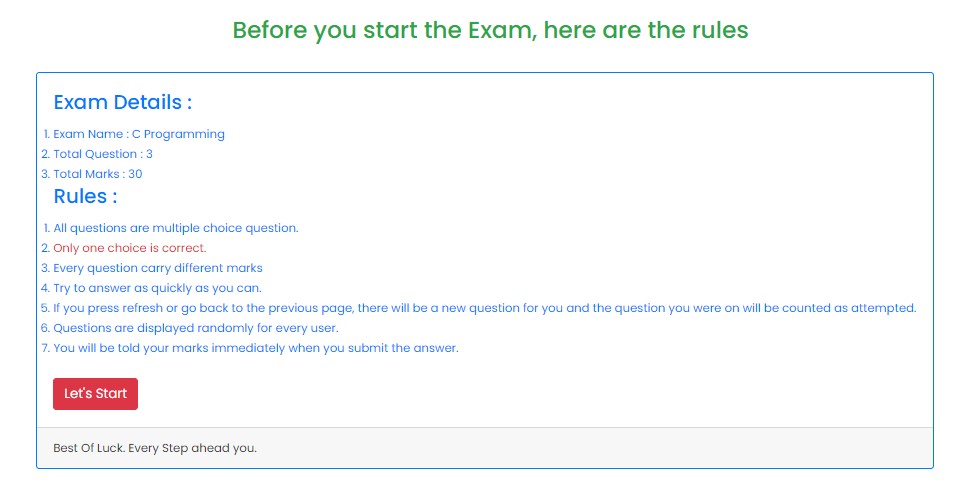
Previous Question Upload: We can upload our offline exam questions or important questions of different courses here.

### 3.Quiz Platform

Our quiz platform serves as a dynamic hub for interactive learning experiences. From pop quizzes to comprehensive assessments, students can test their knowledge, track their progress, and engage in friendly competition with their peers.

F

ig.6 Quiz Rules



F

ig.

7

Quiz

Taking

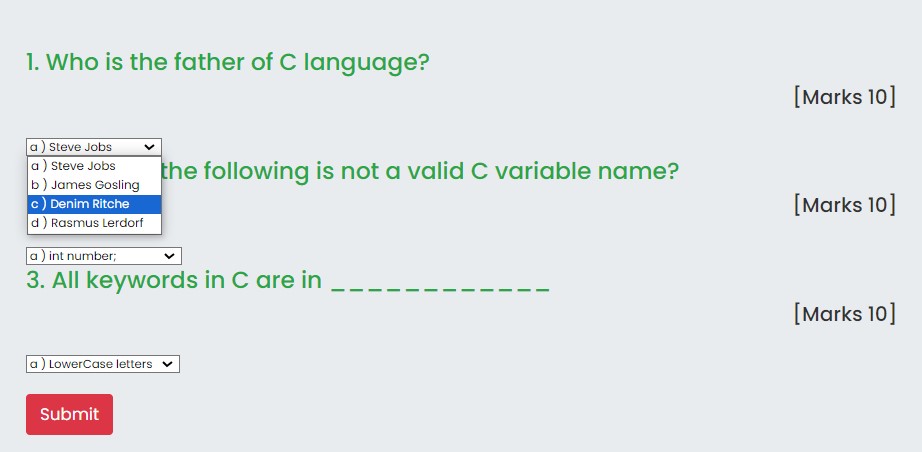


Fig.

8

Check marks

&

Giv

e

ra

ting

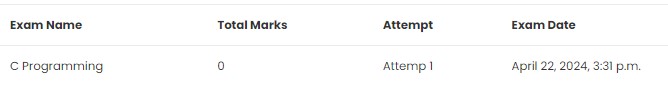
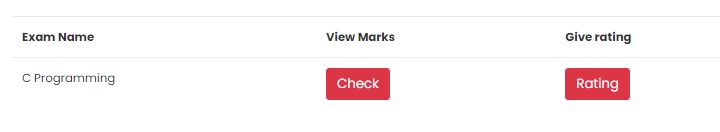


Fig.9 Check marks

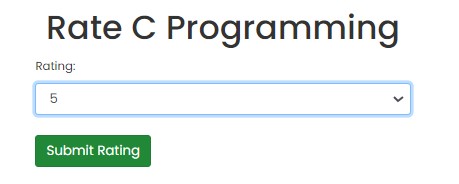


Fig.10 Rate Quiz

### 4.Previous Question Upload

One of the core functionalities of the **AcademiaQuest** platform is the ability for users to upload and access previous questions from exams, quizzes, and assignments. This feature serves as a valuable resource for students, educators, and administrators, enabling them to review past materials, track academic progress, and facilitate more effective teaching and learning experiences.

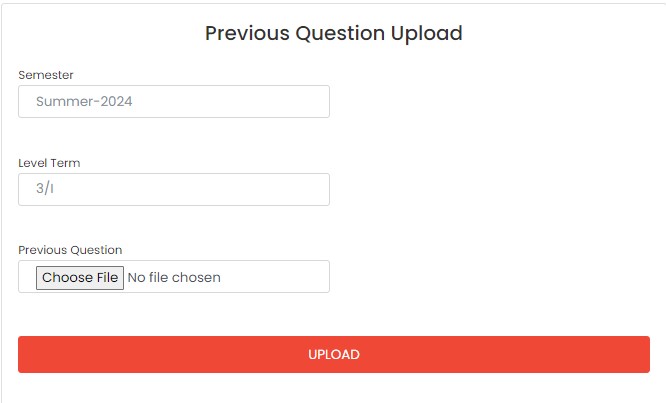


Fig.11 Upload Questions

### 5.Leader Board

The Leaderboard feature in **AcademiaQuest** serves as a dynamic and engaging tool to motivate students, foster healthy competition, and recognize academic achievement. By providing a transparent and interactive platform for tracking student progress and performance, the Leaderboard feature enhances the overall learning experience and promotes a culture of excellence and continuous improvement.

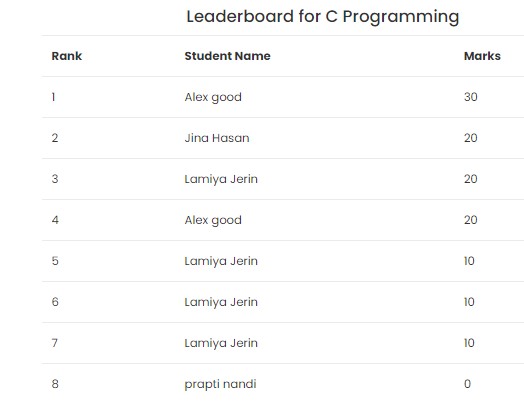
F

ig.

1

2

Ranking



Students may earn achievement badges or virtual rewards for reaching milestones, achieving high scores, or demonstrating exceptional performance. These badges serve as tangible symbols of accomplishment and incentivize continued engagement and participation.

### 6.Study Previous Questions (Question Bank)

The question bank serves as a treasure trove of academic resources, housing a vast collection of past questions across various subjects and disciplines. Students can delve into this repository to review key concepts, reinforce their understanding, and prepare for exams with confidence.

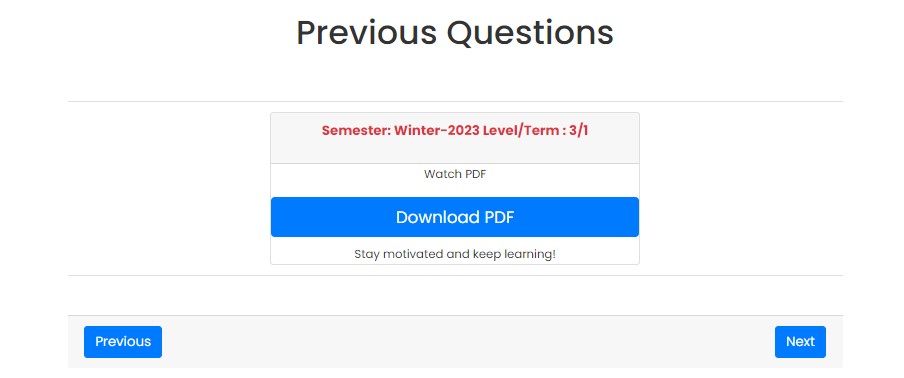
In addition to quizzes and question banks, **AcademiaQuest** offers a curated selection of educational resources to enrich the learning journey. From multimedia tutorials to insightful articles, students can explore diverse learning materials tailored to their interests and learning styles.

Fig.

1

3

Study Previous Questions



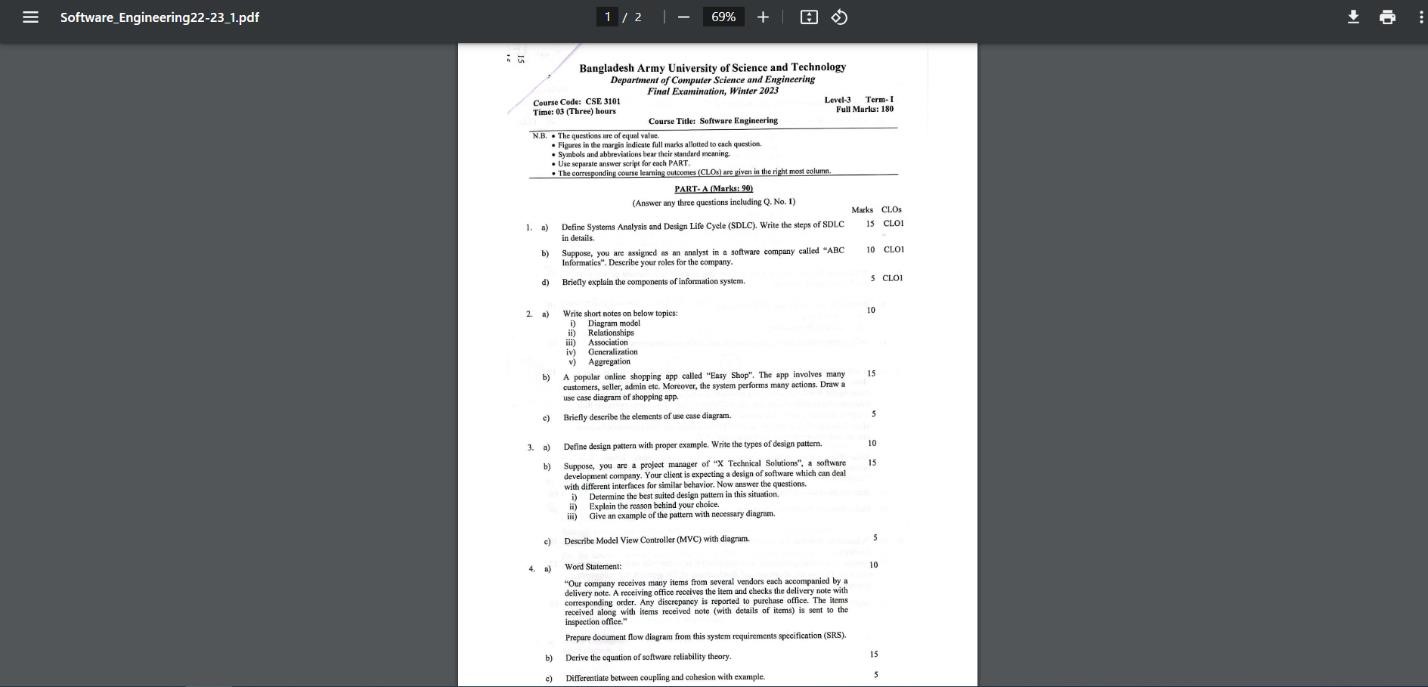
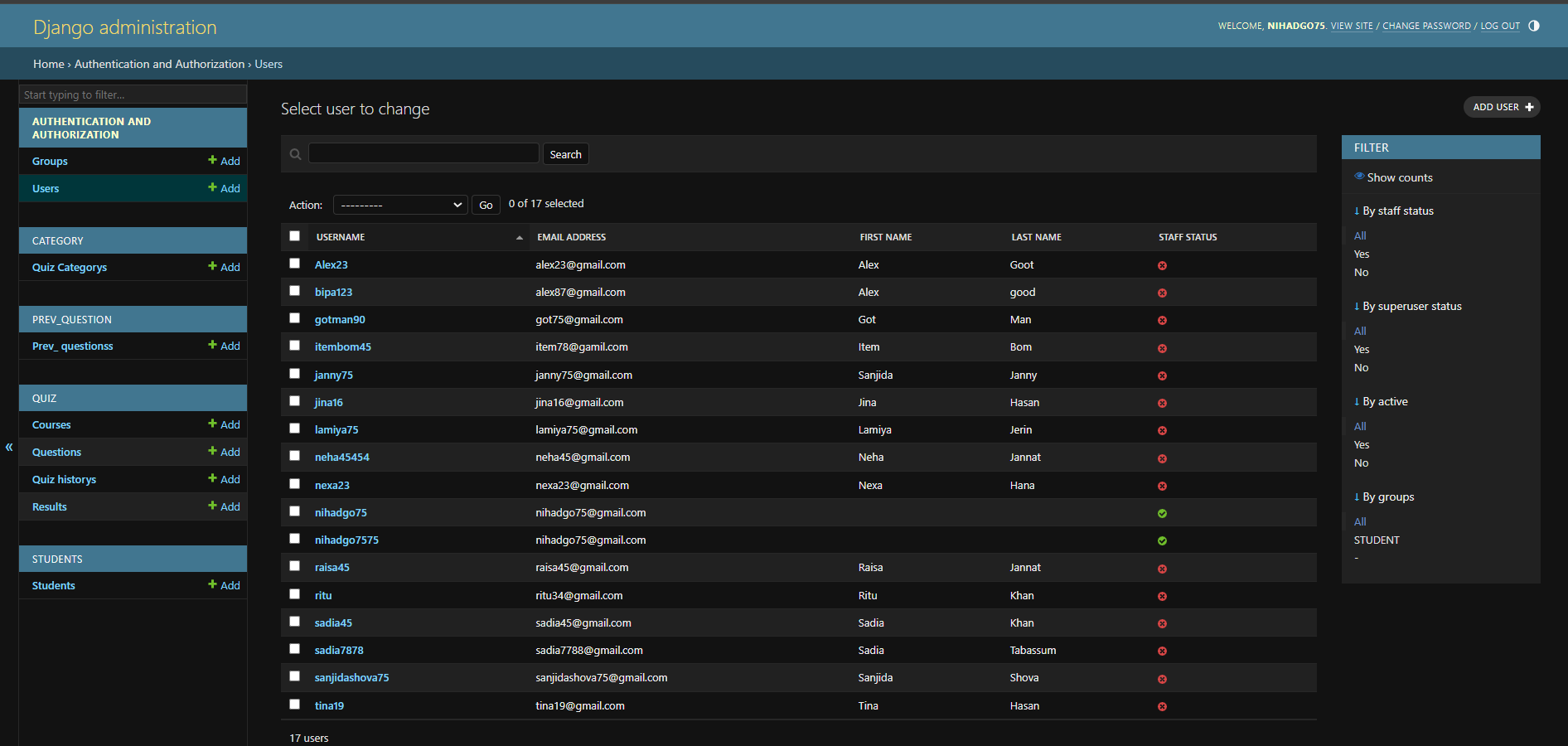
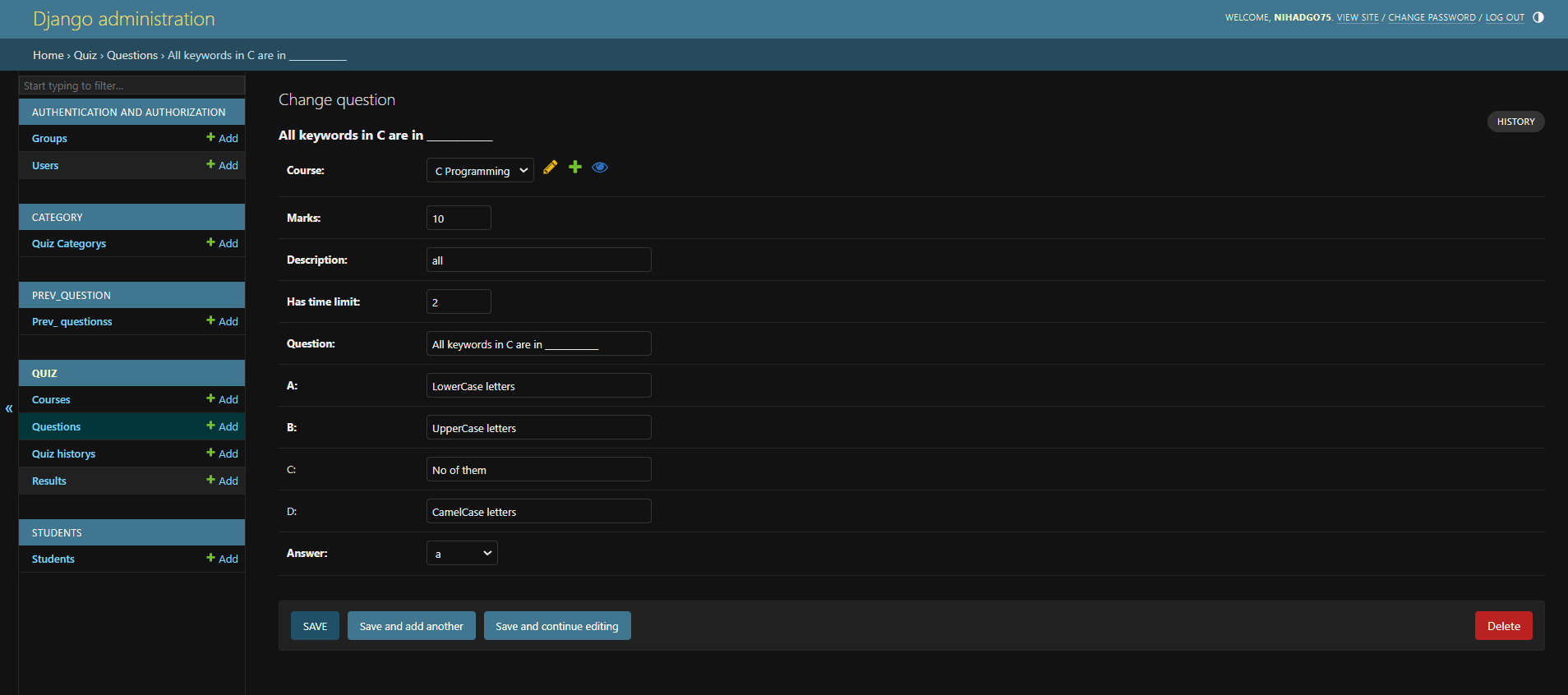


Fig.13 Read Previous Questions

### 7.Admin Panel (Administration)

The admin panel in a quiz system is like a control center for administrators. They can manage users, quizzes, student performance, reports, and system settings, ensuring smooth operation and valuable insights.





# 

# Chapter 3: Different Principles Used in the Project

In the development of **AcademiaQuest,** we meticulously adhered to various software design principles to ensure the robustness, scalability, and maintainability of the platform. Let's delve into each principle with examples of how they were applied: **3.1 Single Responsibility Principle (SRP)**

The Single Responsibility Principle states that a class should have only one reason to change, meaning it should have only one responsibility or job within the system. This principle promotes clarity, cohesion, and ease of maintenance in the codebase.

*Example:* In **AcademiaQuest,** we have separate classes responsible for specific tasks such as user authentication, quiz management, and question retrieval. For instance, the **UserAuthenticationService** class handles user authentication logic, while the **QuizManager** class manages quiz creation, retrieval, and grading. Each class is focused on a single responsibility, making the codebase more modular and easier to understand and modify.

### 3.2 Open/Closed Principle (OCP)

The Open/Closed Principle states that software entities (classes, modules, functions, etc.) should be open for extension but closed for modification. In other words, existing code should be able to be extended with new functionality without altering its source code.

*Example:* In **AcademiaQuest,** we achieved the Open/Closed Principle by using inheritance and polymorphism. For instance, when introducing new types of quizzes, instead of modifying the existing **Quiz** class, we create subclasses such as **MultipleChoiceQuiz** and **TrueFalseQuiz** that extend the base **Quiz** class. This allows us to add new quiz types without changing the existing codebase, promoting code reuse and maintainability.

### 3.3 Liskov Substitution Principle (LSP)

The Liskov Substitution Principle states that objects of a superclass should be replaceable with objects of its subclass without affecting the correctness of the program. In other words, subclasses should be substitutable for their base classes without altering the behavior of the program.

*Example:* In **AcademiaQuest,** we adhere to the Liskov Substitution Principle by ensuring that subclasses can be used interchangeably with their base classes. For instance, when retrieving questions from the question bank, we can use polymorphism to treat all question types (e.g., multiple-choice, true/false) uniformly. This allows us to seamlessly switch between different question types without needing to modify the code that consumes them.

### 3.4 Interface Segregation Principle (ISP)

The Interface Segregation Principle states that clients should not be forced to depend on interfaces they do not use. It advocates for the creation of specific, client-focused interfaces rather than large, monolithic interfaces.

*Example:* In **AcademiaQuest,** we design interfaces tailored to specific client requirements, promoting modularity and flexibility. For example, instead of having a single **IQuestion** interface with methods for all types of questions, we create separate interfaces like **IMultipleChoiceQuestion** and **ITrueFalseQuestion**, each containing methods relevant to its specific question type. This allows clients to depend only on the interfaces they need, reducing unnecessary dependencies and promoting code cohesion.

### 3.5 Dependency Inversion Principle (DIP)

The Dependency Inversion Principle states that high-level modules should not depend on lowlevel modules but should depend on abstractions. It advocates for decoupling high-level modules from the implementation details of low-level modules, promoting flexibility and testability.

*Example:* In **AcademiaQuest,** we apply the Dependency Inversion Principle by using dependency injection to invert the dependencies between high-level and low-level modules. For example, instead of directly instantiating database access objects in the **QuizManager** class, we inject them as dependencies via constructor injection. This allows us to easily substitute different database implementations or mock objects for testing purposes without modifying the **QuizManager** class, enhancing modularity and testability.

By diligently applying these principles throughout the development of **AcademiaQuest,** we ensure that the platform is well-designed, maintainable, and adaptable to future changes and enhancements. Each principle contributes to the overall quality and reliability of the software, fostering a codebase that is robust, scalable, and resilient to change.

# 

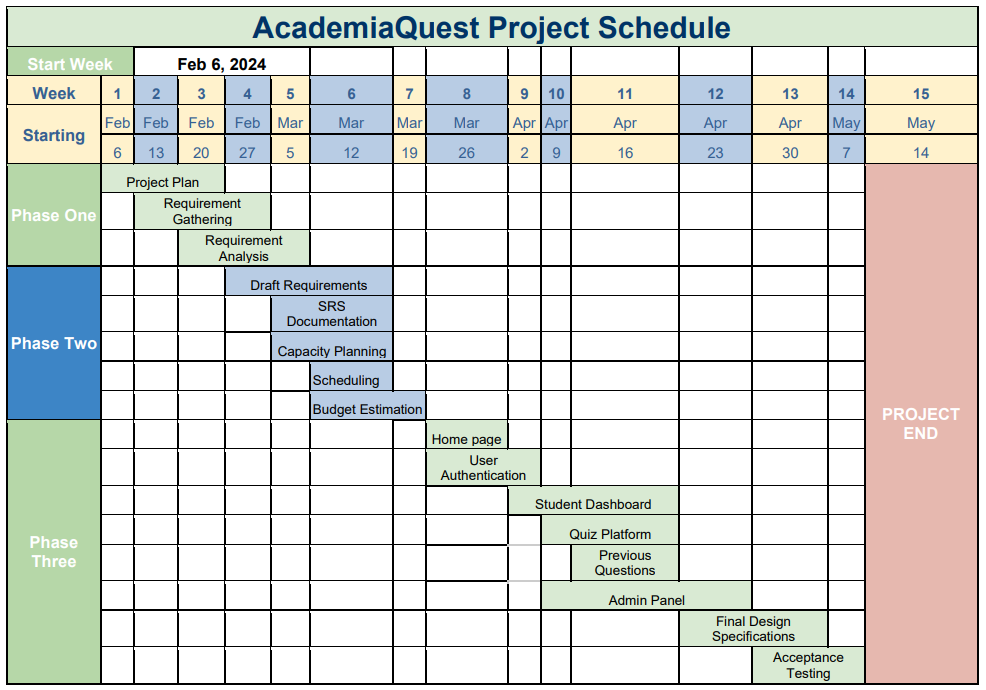
# Chapter 4: Project Time and Budge Management

Effective time and budget management are critical components of project success. Throughout the development lifecycle of **AcademiaQuest**, we meticulously planned and monitored our resources to ensure optimal utilization and adherence to project timelines.

The following table provides an overview of our planned and actual time allocation and budget expenditure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **AcademiaQuest Project Budget Plan** | | | | | |
| Requirement | Feature Describe | By (Team member) | Man Hour | Hourly Charge in $ | Budget in $ |
| 1 | Home page | NH | 15 | 20 | 300 |
| 2 | login &  Registration | NH | 20 | 30 | 600 |
| 3 | Student Dashboard | NH | 30 | 20 | 600 |
| 4 | Admin panel | NH | 30 | 30 | 900 |
| 5 | Courses | NH | 10 | 25 | 250 |
| 6 | Quiz platform | NH | 40 | 30 | 1200 |
| 7 | Marks Calculation | NH | 20 | 30 | 600 |
| 8 | Leaderboard | NH | 20 | 25 | 500 |
| 9 | Previous Question | NH | 30 | 30 | 900 |
| 10 | Quiz History | NH | 20 | 25 | 500 |
|  |  | Total Man Hour = 235 H | | Total Budget = 6350 $ | |

Fig.14 Budget Plan



# 

# Chapter 5: Conclusion

In conclusion, AcademiaQuest represents a testament to the power of innovation, collaboration, and commitment to excellence. By harnessing the latest advancements in technology and adhering to the principles of software engineering, we have created a platform that transcends the boundaries of traditional education.

As we embark on this transformative journey, we remain steadfast in our dedication to empowering learners, enriching educational experiences, and shaping the future of learning. With AcademiaQuest, the quest for knowledge knows no bounds, and the pursuit of excellence becomes an exhilarating adventure.