CSD 416 PROJECT PHASE II PRESENTATION

Autoquestgen

Group-8

Group Members

20CSB49	MDL20CS101	Sabeel EK
20CSB35	MDL20CS069	Mahesh Krishna
20CSB51	MDL20CS0103	Sangeeth TS
20CSB42	MDL20CS091	Pranav K Pradeep

Project Guide

Mrs.Sindhu L

Department of Computer Science and Engineering
Model Engineering College, Thrikkakara

Contents

Introduction

AutoQuestGen is a platform to minimize the efforts required to conduct an exam.

- Create an exam under 10 minutes.
- Quick question creation using NLP
- Automatic evaluation.
- Instant results.

Problem Statement

- Conducting an offline exam is tedious for the teachers, as it is a time-consuming process because it involves setting up question papers and then it's evaluation.

Proposed System

"AutoQuestGen" is an online question generation and exam platform that allows teachers to easily generate Multiple Choice Questions for an exam and then students can attend these exams and get their marks.

Advantages

- Generate MCQs easily.
- Enchance Teaching Experience for Teachers.
- Quickly Create Exams.
- Get Exam Results ASAP.

Software & Hardware Requirements

Operating Environment

Browser with internet connection.

Hardware Interface

Personal Computer.

Software Interface

WebApp Platform.

Tools

- Design: Figma
- Frontend: HTML, CSS, JavaScript, Bootstrap
- Backend: Django
- Database: SQLite

Literature Survey Summary

Name of Paper (Year)	Name of Author(s)	Methodology	Advantages	Disadvantages
Automatic Question Paper Generator System (2020)	Dubey Harish, Tamore Hardik, Padhi Sagar, Prof. Manisha Bharambe	Automated template generation, Literature review	Streamlined question paper creation	Focus on traditional testing formats
Question Paper Generator and Answer Verifier (2017)	Prateek Pisat, Shrimangal Rewagad, Devansh Modi, Ganesh Sawant	Digital assessment platform, Multiple question formats	Efficient grading, Diverse question types	Limited to digital platforms
Fuzzy Logic Based Intelligent Question Paper Generator (2014)	Suraj Kamya, Madhuri Sachdeva, Navdeep Dhaliwal, Sonit Singh	Fuzzy logic for parameter selection	Autonomous operation, Reduced manual effort	Complexity in parameter tuning
Secure Automatic Question Paper with Reconfigurable Constraints (2021)	I. Ragasudha, M. Saravanan	Python-based system, Bloomâs Taxonomy	Consistency in question quality, Automated process	Programming knowledge required
Automatic Generation of Question Paper from User Entered Specifications (2016)	Gauri Nalawade, Rekha Ramesh	Semantically tagged question repository	Tailored question papers, XML format output	Setup complexity of semantic tagging

Literature Survey Summary

Name of Paper (Year)	Name of Author(s)	Methodology	Advantages	Disadvantages
Automated Exam Question Generator using Genetic Algorithm (2017)	Tengku Nurulhuda Tengku Abd Rahim, Zalilah Abd Aziz, Rose Hafsah Ab Rauf, Noratikah Shamsudin	Genetic algorithm, Bloomâs Taxonomy	High adaptability, Customized difficulty levels	Initial setup and data requirements
Automatic Question Generation With Classification Based On Mind Map (2019)	Selvia Ferdiana Kusuma, Daniel Oranova Siahaan, Chastine Fatichah, Mohammad Farid Naufal	Mind mapping for question generation	Innovative use of mind maps, Varied difficulty levels	Specific to Indonesian educational system
Subjective Answers Evaluation Using Machine Learning and Natural Language Processing (2021)	Muhammad Farrukh Bashir, Hamza Arshad, Abdul Rehman Javed, Natalia Kryvinska, Shahab S. Band	ML and NLP for answer evaluation	Enhanced accuracy, Reduced subjectivity	High technical expertise required
Efficient Automated Processing of Unstructured Documents Using Artificial Intelligence (2021)	Dipali Baviskar, Swati Ahirrao, Vidyasagar Potdar, Ketan Kotecha	AI for processing unstructured data	Improved data processing, AI-driven insights	Implementation complexity, High resource demand
Online Subjective Answer Verifying System Using Artificial Intelligence (2020)	G. Jagadamba, Chaya Shree G.	AI techniques for answer verification	Speed in evaluation, Consistent analysis	Dependency on AI accuracy and data quality

Software Requirements Specification

Purpose

This document outlines the functional and non-functional requirements of AutoQuestGen, detailing the features essential for a platform that helps teachers to create question papers and for students to attend these exams.

Project Scope

AutoQuestGen should help teachers conduct exams quickly without much hassle.

Product Functions

The core functions involve :

- **Question Generation using NLP:** Quick generation of an MCQ question from a text.
- **Exam Platform:** A platform where the students can attend exams available to them.
- **Educator Interface:** Offers educators an interface to specify criteria for question paper generation, preview generated question papers, and make adjustments as needed.

User Classes and Characteristics

User classes include:

- Teacher
- Student

Design and Implementation Constraints

- Generate questions at high speeds.
- Input text should be considerably large.

User Documentation

- Prompts the teachers to enter input text and then generates questions from it.

Assumptions

The project assumes that the users:

- have a compatible computer system.
- have a stable internet connection.

Hardware and Software Requirements

Hardware Requirements

1. For optimal performance, you need at least 2 GB of RAM.
2. A computer with sufficient processing power is needed.
3. Any browser compatible device.

Software Requirements

- Software requirements for development process:
  - Python
  - Django
  - Visual Studio Code

Functional Requirements

Question Generation Using NLP

- This feature enables the automatic generation of diverse question papers using NLP.
- Teachers set criteria (e.g., subject, Difficulty level), system generates unbiased question paper from training data.

Admin Interface for Question Management

- Admins can add, categorize, and manage questions through an intuitive interface.
- Admin can approve teachers to use AutoQuestGen

Non-Functional Requirements

Performance Requirements

- The system should generate question papers within a reasonable time frame, even for large question databases and complex criteria.

Safety Requirements

- Database backup required to prevent data loss due to crashes or failures.

Usability

- The userinterface is intuitive and user-friendly, allowing teachers to easily specify criteria and navigate through the system.

SOFTWARE DESIGN DOCUMENT

Purpose

The purpose of this document is to present the design implementations to implement AutoQuestGen.

System Architecture

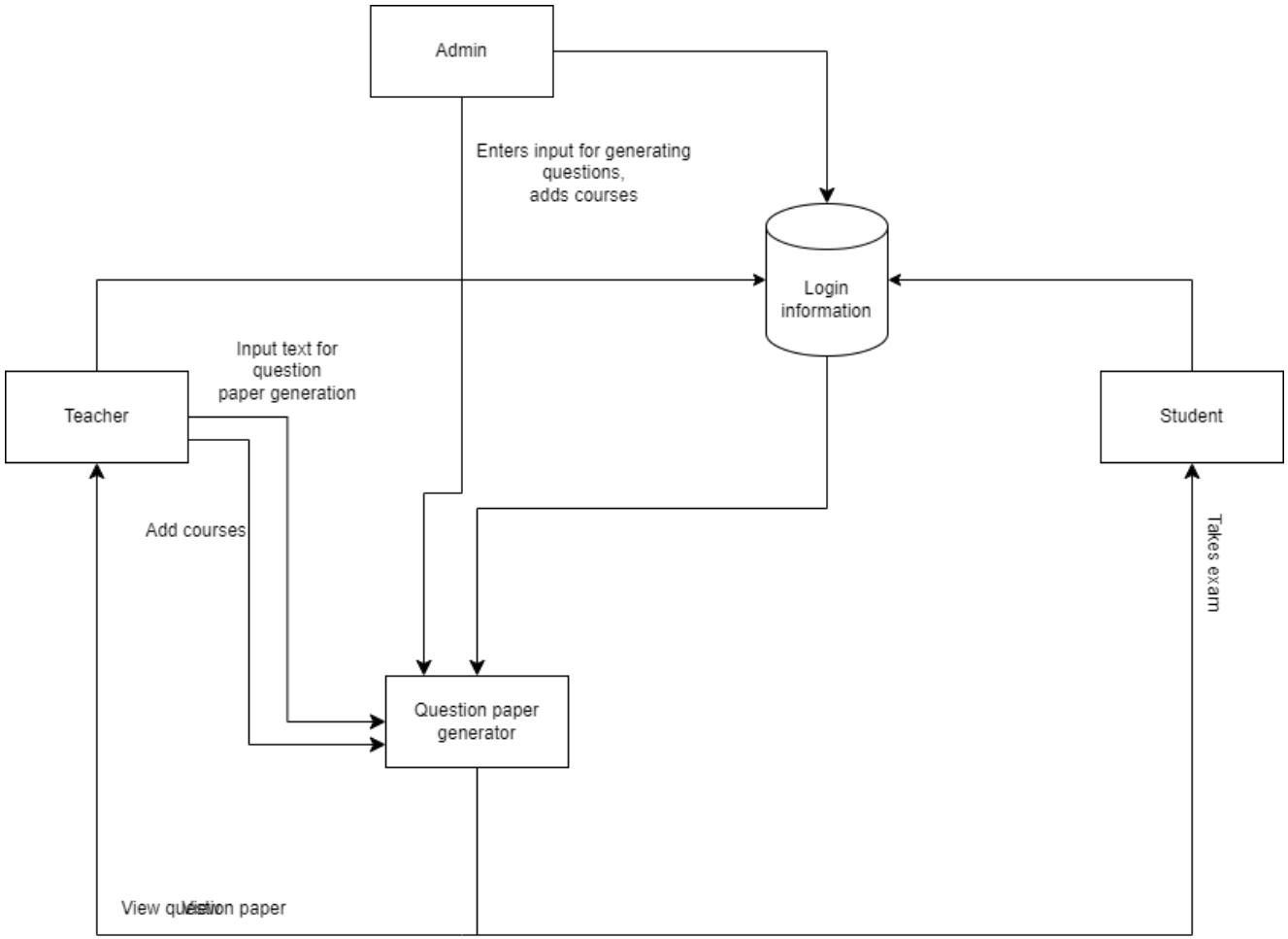

Figure 1: Architecture diagram

Architecture Diagram

1. **User Authentication Module:**
   - Provides a user-friendly interface for seamless interaction with the application.
2. **Question Generation Module:**
   - Focuses on generating the questions for exams based on the input text.
3. **Exam Platform Module:**
   - The students can attend the exams and view their marks.

Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the data flow through a system modeling its process aspects.

DFD Level 0

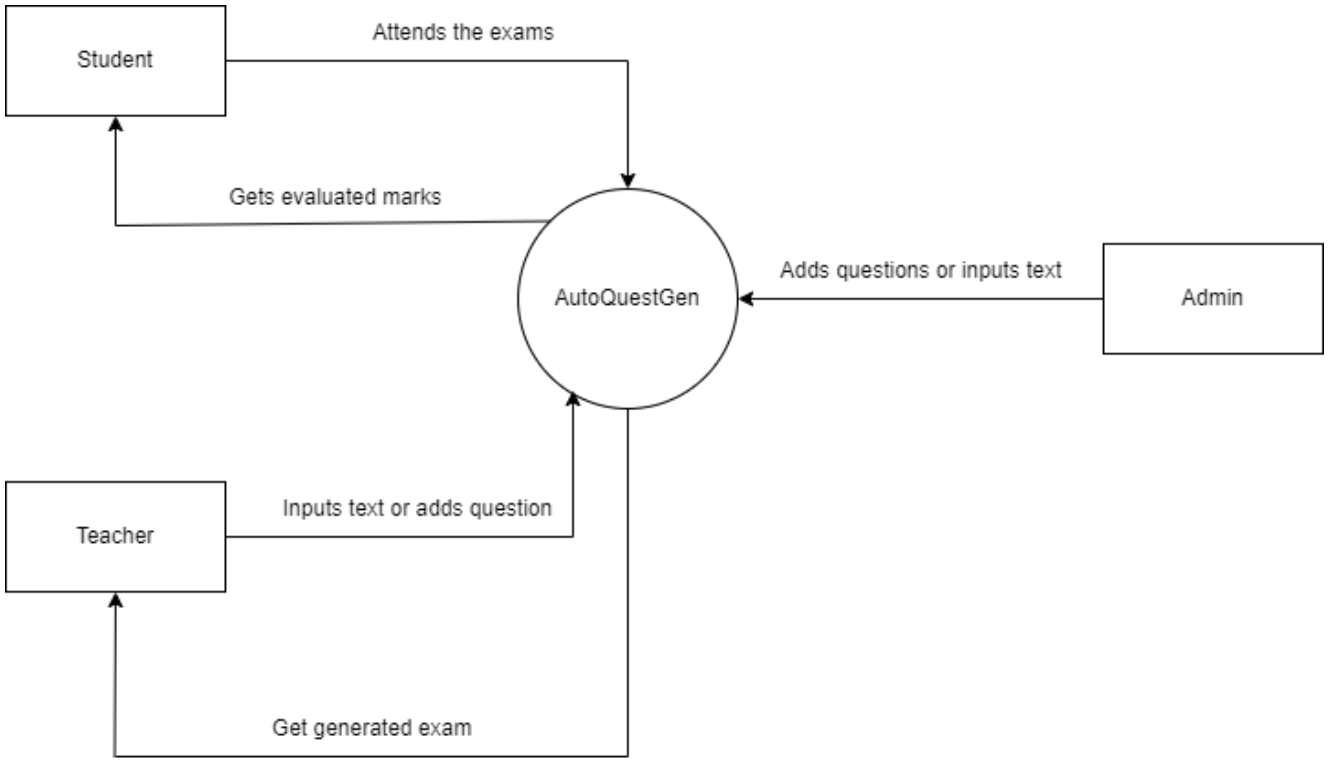

Figure 2: DFD Level 0

DFD Level 1

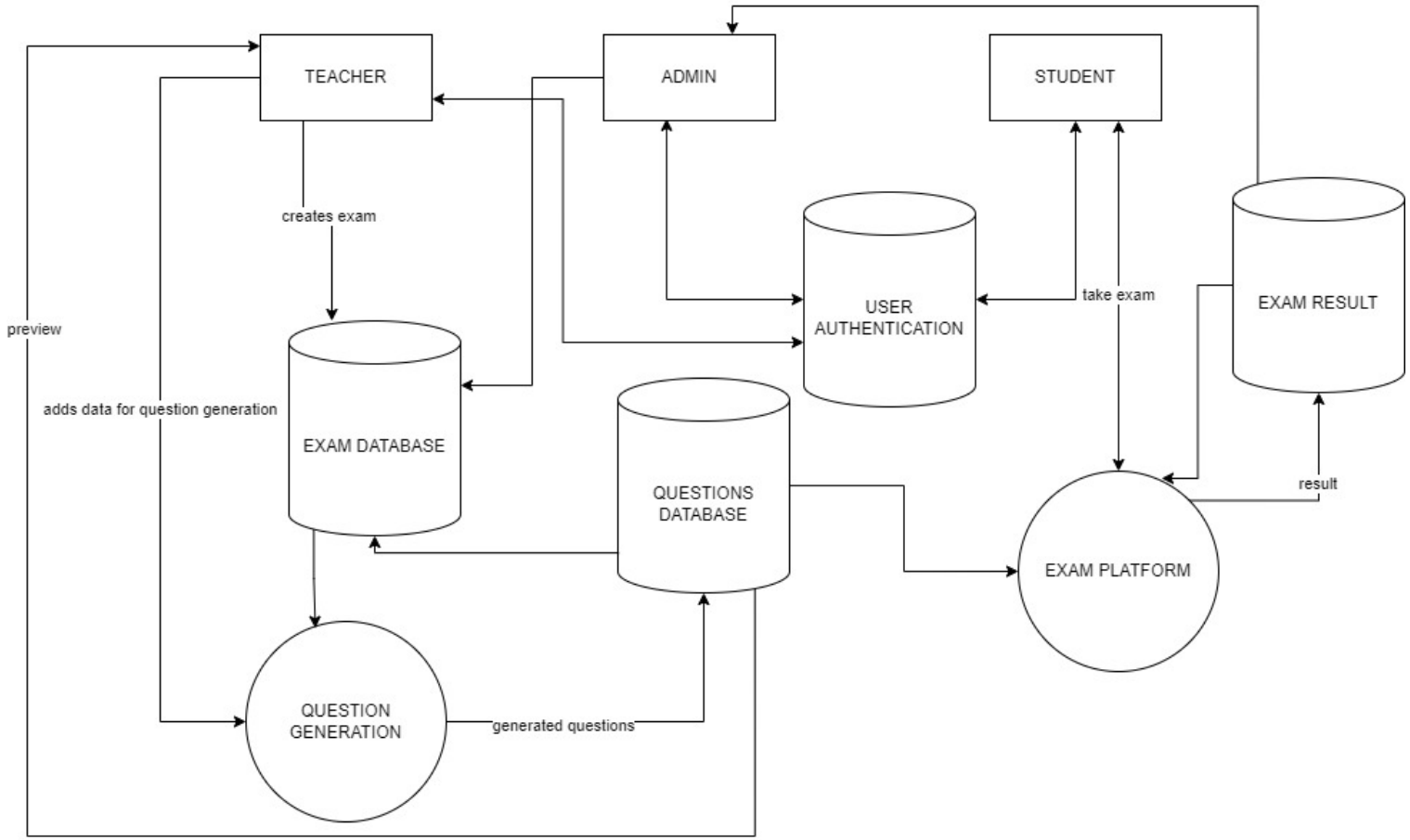

Figure 3: DFD Level 1

Use Case Diagram

The use case diagram outlines how users interact with the system.

Use Case Diagram

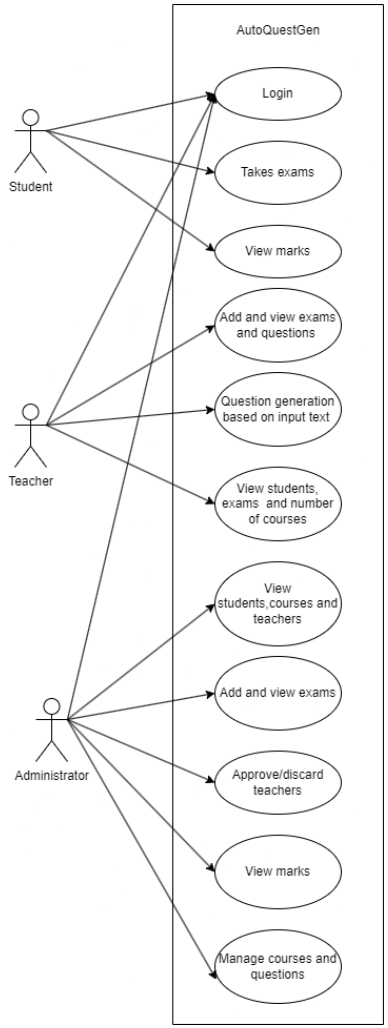

Figure 4: Use Case Diagram

Activity Diagram

The activity diagram visually represents the flow of actions illustrating the sequential steps.

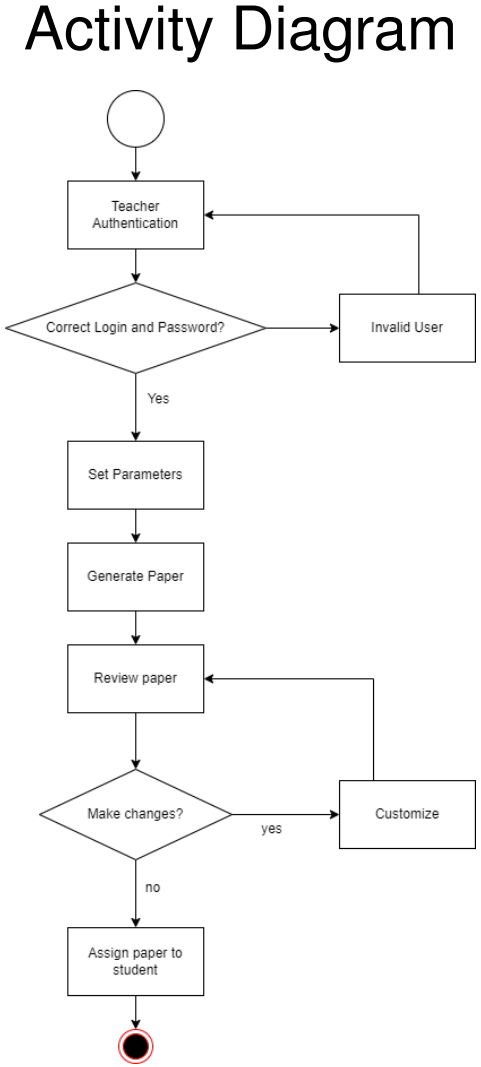

Figure 5: Teacher

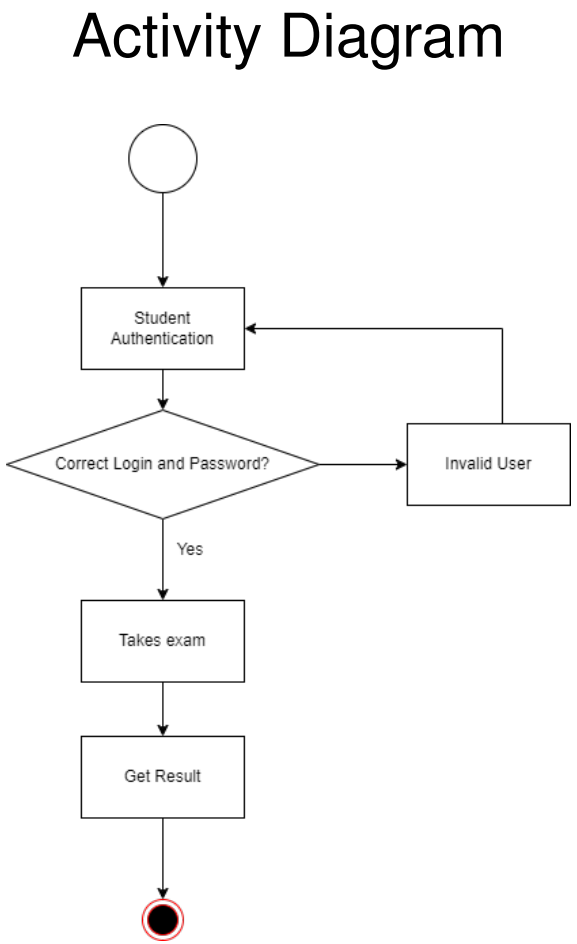

Figure 6: Student

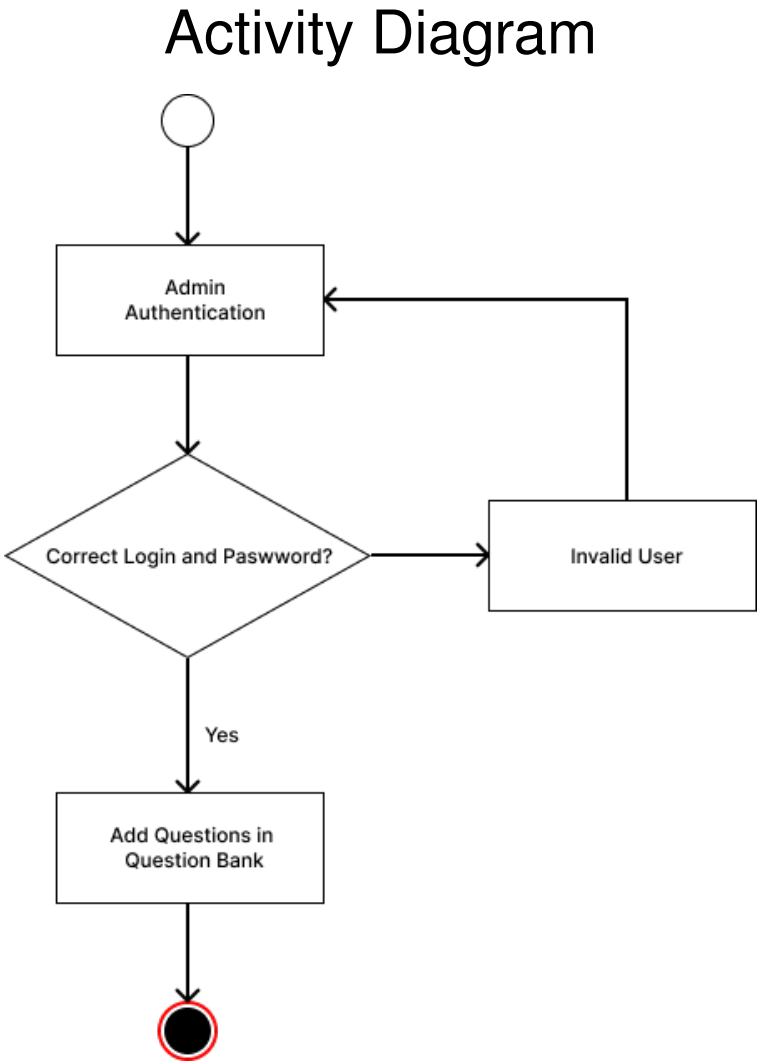

Figure 7: Admin

Implementation

Algorithm for Creating an exam

1. Start
2. Initialization and authentication
   - Creates new teacher account
   - Account is validated by admin
3. Add Exam: Teacher can create a new exam
   - Determine number of questions, total marks,etc
   - Add exam to database
4. Generate questions/Add question
   - Input the text to generate the questions from
   - Generate the question
   - Set the marks, exam and other criteria.
5. Add the questions to the database
6. View the questions and edit them based on preference
7. Stop.

Algorithm for Question generation

1. Start
2. Teacher inputs the text for question generation
3. The input text is summarized using Bert
4. Keyword Extraction using RAKE from summarized content
5. Generate incorrect options using Sense2Vec
6. Question Generation using T5 Transformer Model
7. Stop.

Algorithm for Attending Exam

1. Start
2. Initialization and authentication
   - Creates new user account
3. Start Exam: The Student can start attempting the exam
   - View the available exams
   - Start the exam
4. View Marks: Student can view their marks.
5. Stop.

Developer Tools

- Django: Django is a high-level Python web framework used for rapid development and clean, pragmatic design. It provided the foundation for building the backend logic and web application structure of AutoQuestGen.
- HTML: HTML(Hypertext Markup Language) is used for creating the structure and content of web pages within the AutoQuestGen application.
- CSS: CSS (Cascading Style Sheets) is used for styling the HTML elements, defining the layout, and enhancing the visual presentation of the AutoQuestGen user interface.
- Python: Python programming language was extensively used for implementing the backend logic, algorithms, and business logic of AutoQuestGen

Testing

Testing

- Software testing involves the execution of a software component or system component to evaluate one or more properties of interest.
- Test techniques include the process of executing a program or application with the intent of finding software bugs, and to verify that the software product is fit for use.

Testing Methodologies

- Software Testing is a method to check whether the actual software product matches expected Requirements and ensuring software products are defect-free involve the execution of software components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps, or missing requirements in contrast to actual requirements.

Unit Testing

1. User Authentication Module

Steps :

- Test if user login is successful only with correct credentials.
- Test if access tokens created during authentication is valid.
  Expected Output :
- Successful login is achieved only with correct credentials.
- Valid access tokens are generated during authentication.

Status :

- Pass

2. Question Generation Module

Steps :

- Test if the question is generated successfully.
- Test if correct options are generated.
- Test if questions are added to the correct exam.

Expected Output :

- Question is generated and added to the database.
- Options are generated and correct answer is selected.

Status :

- Pass

3. Exam Module

Steps :

- Test if the questions are added to the database.
- Test if the student can attend the exam.

Expected Output :

- Questions are visible on the platform.
- Multiple Students can access the questions.
- Students can view their marks

Status :

- Pass

Integration Testing

Integration testing ensures that individual software modules work together as expected when combined and tested as a group.

1. **Integration between User Interface and Backend:**
   Test the integration between the frontend user interface and the backend server to ensure seamless communication and data exchange
2. **Integration between Database and Application Logic:**
   Test the integration between the database and application logic to verify data consistency and accuracy.

System Testing

Combine and test all components of the application.
Steps:

- Test the entire system workflow from user registration to displaying marks to ensure all components work together seamlessly.

Expected Output:

- Ensure that all components work seamlessly together without introducing new issues.

Status:

- Pass.

Graphical User Interface

Landing Page

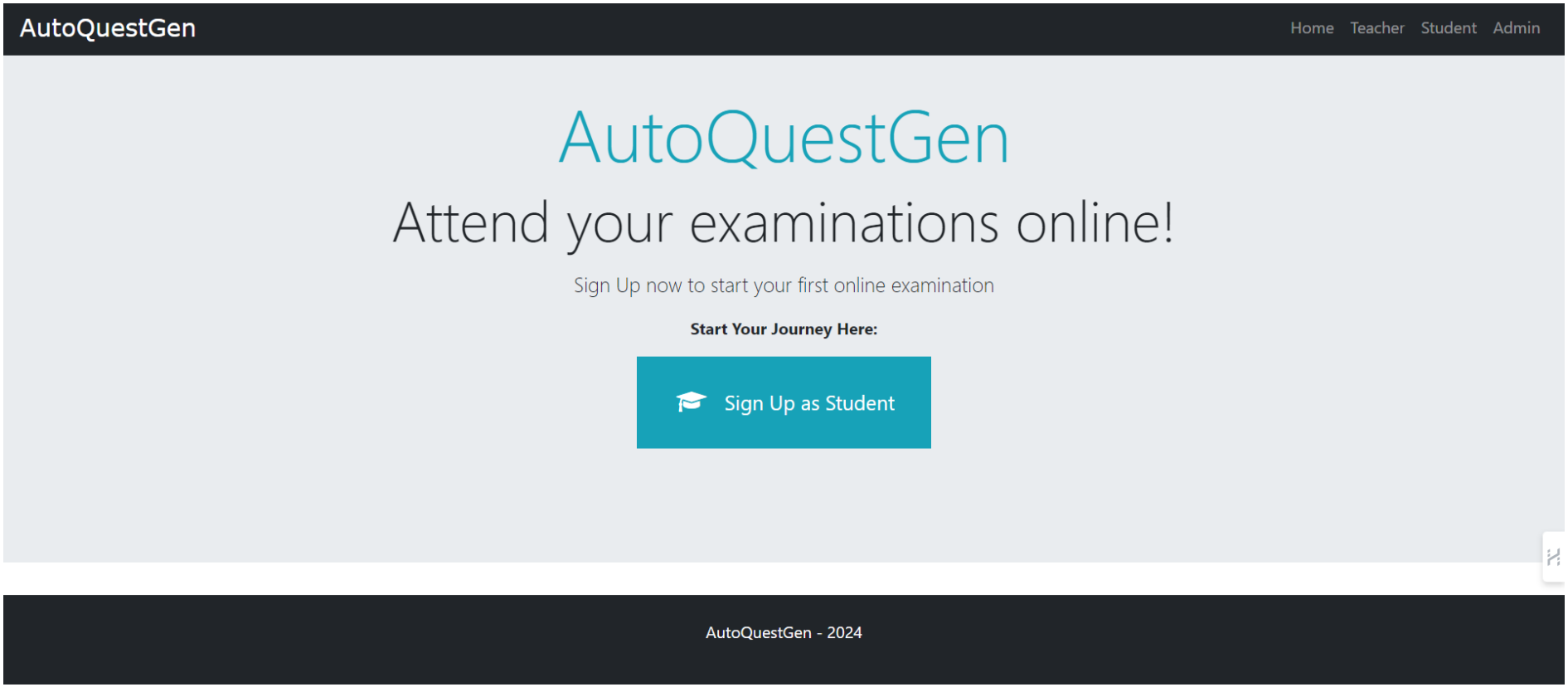

Figure 8: Landing Page

Manage Exams

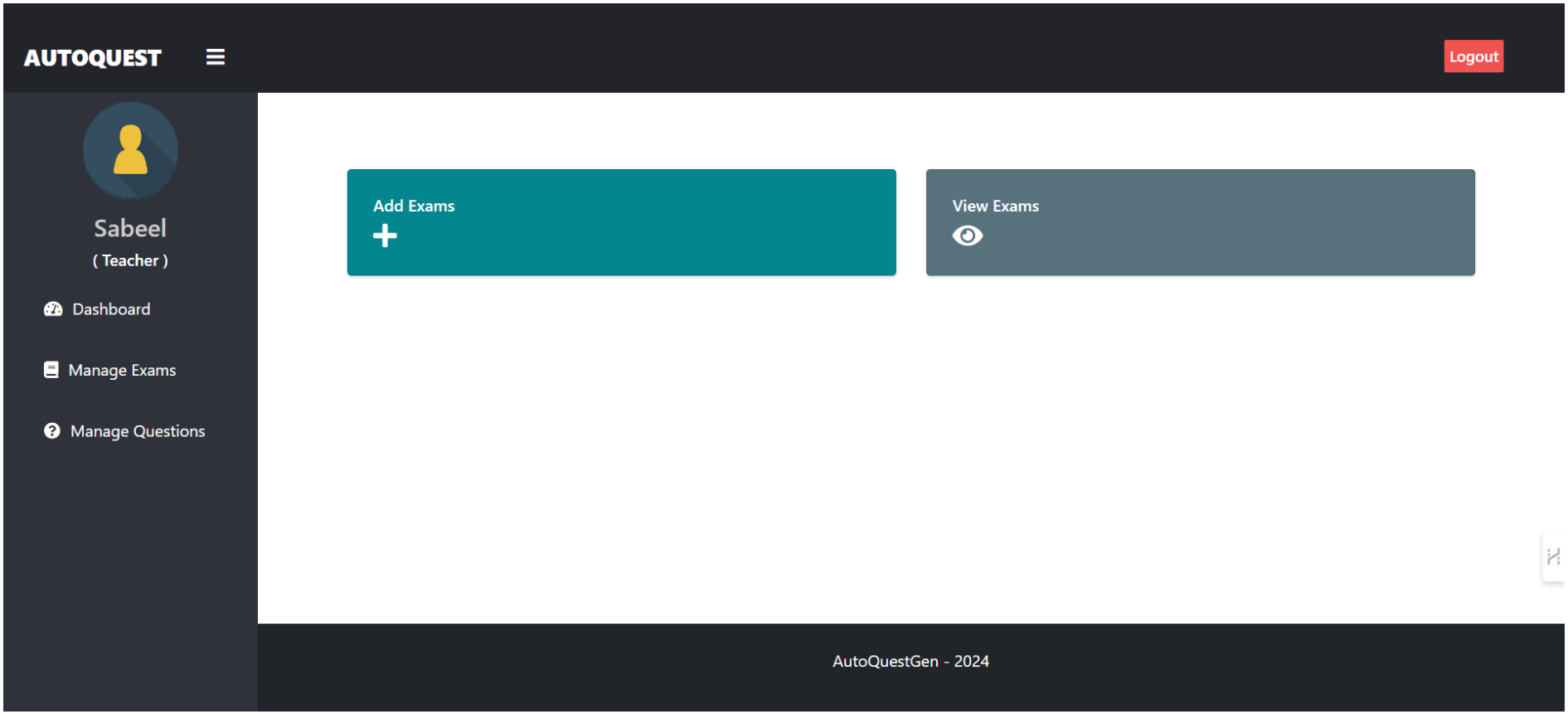

Figure 9: Manage Exams

Add Exam

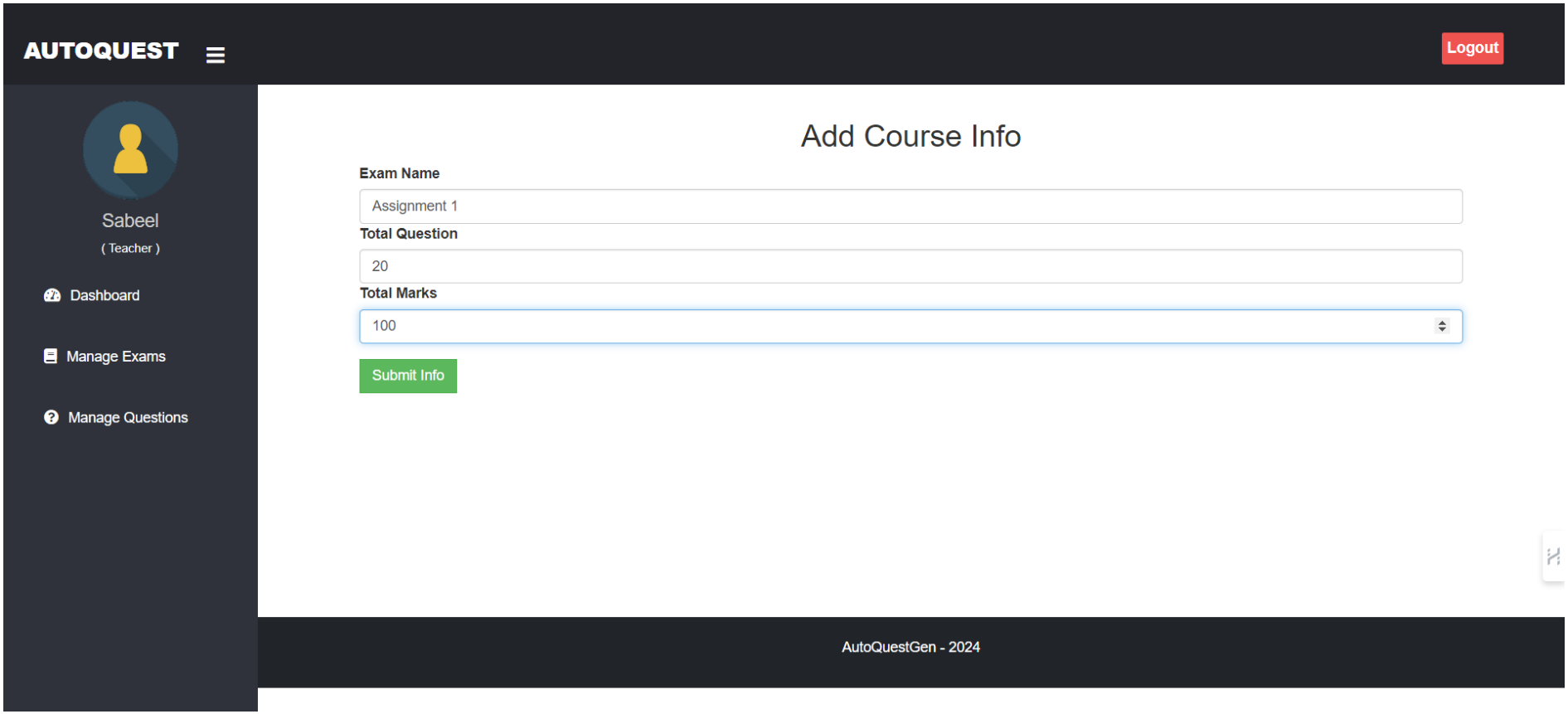

Figure 10: Add Exam

View Exams

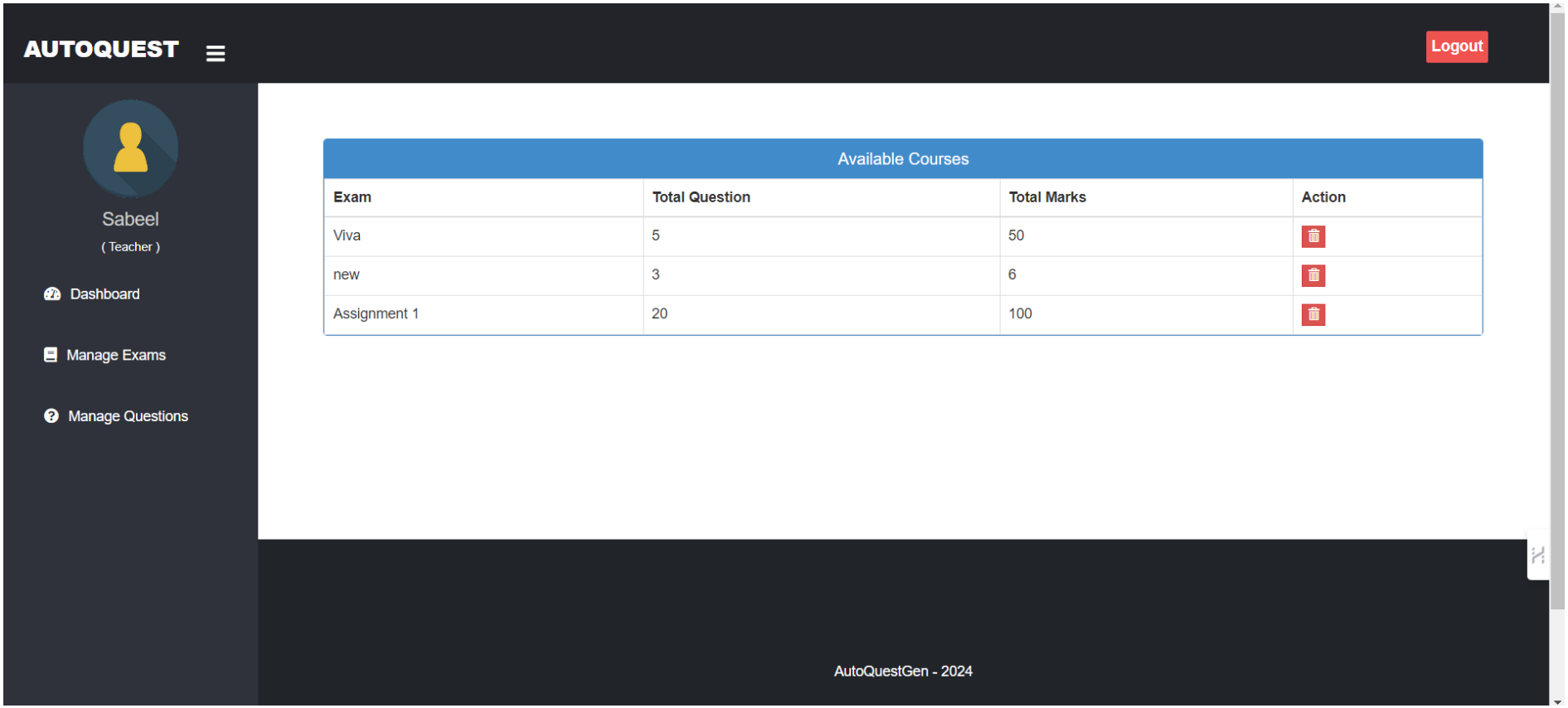

Figure 11: View Exams

Manage Questions

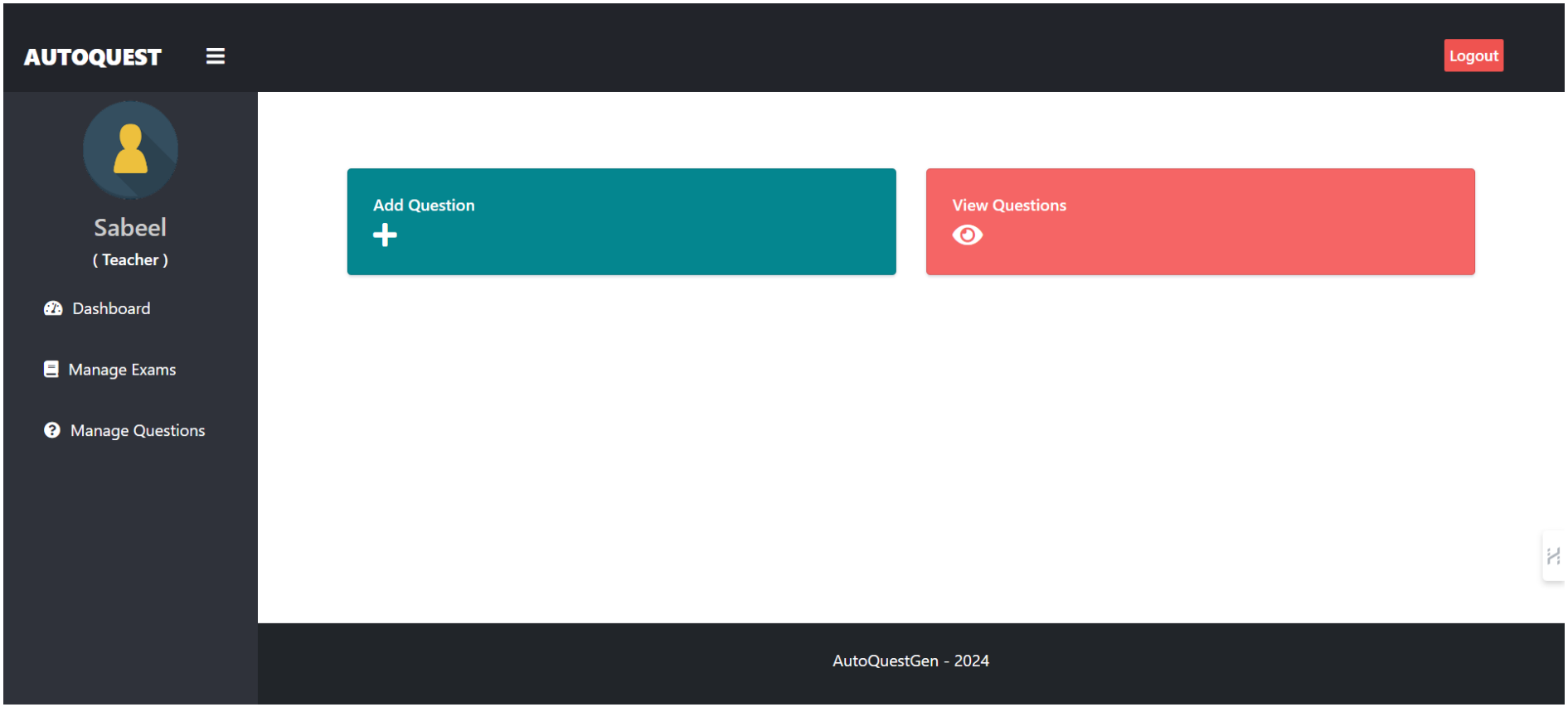

Figure 12: Manage Questions

Add/Generate Questions

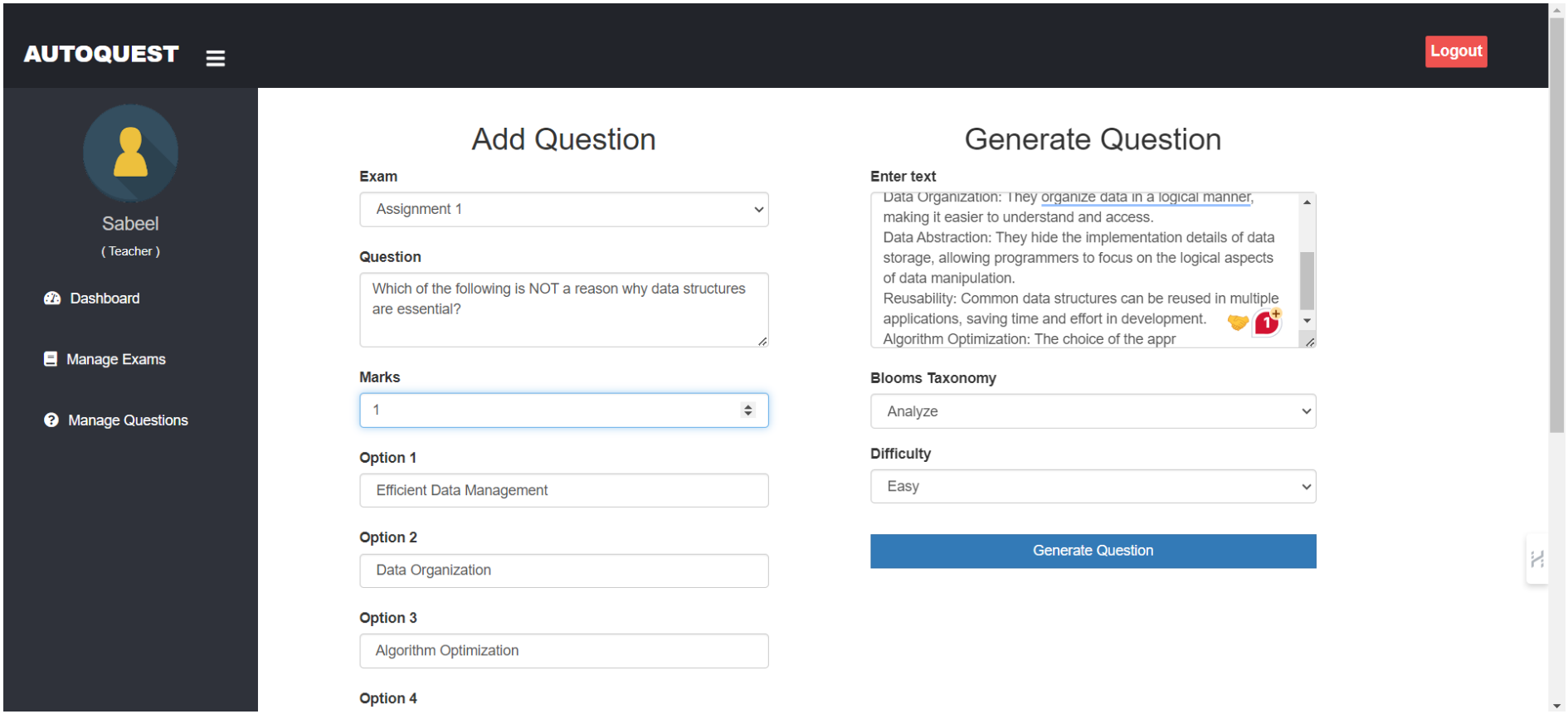

Figure 13: Add/Generate Questions

View Questions

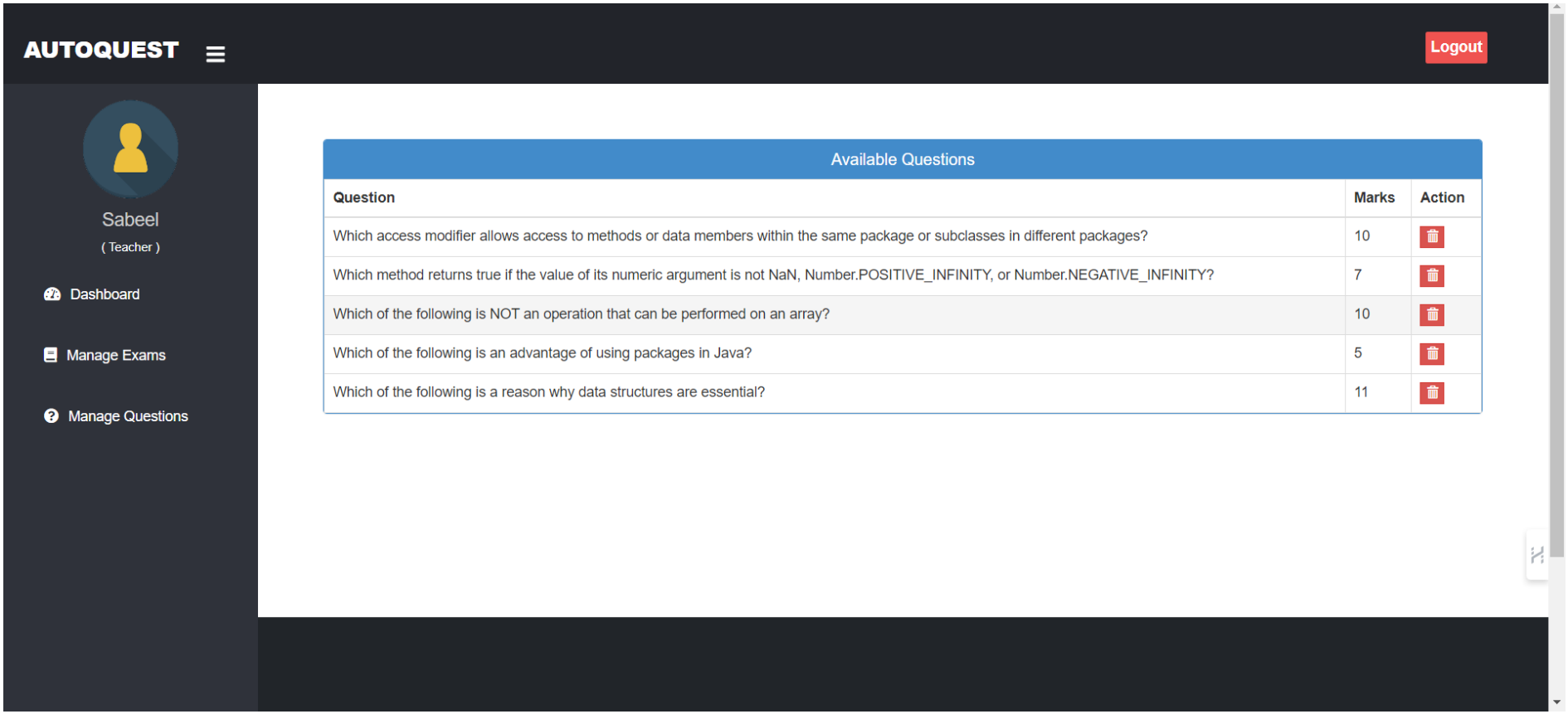

Figure 14: View Questions

Student Dashboard

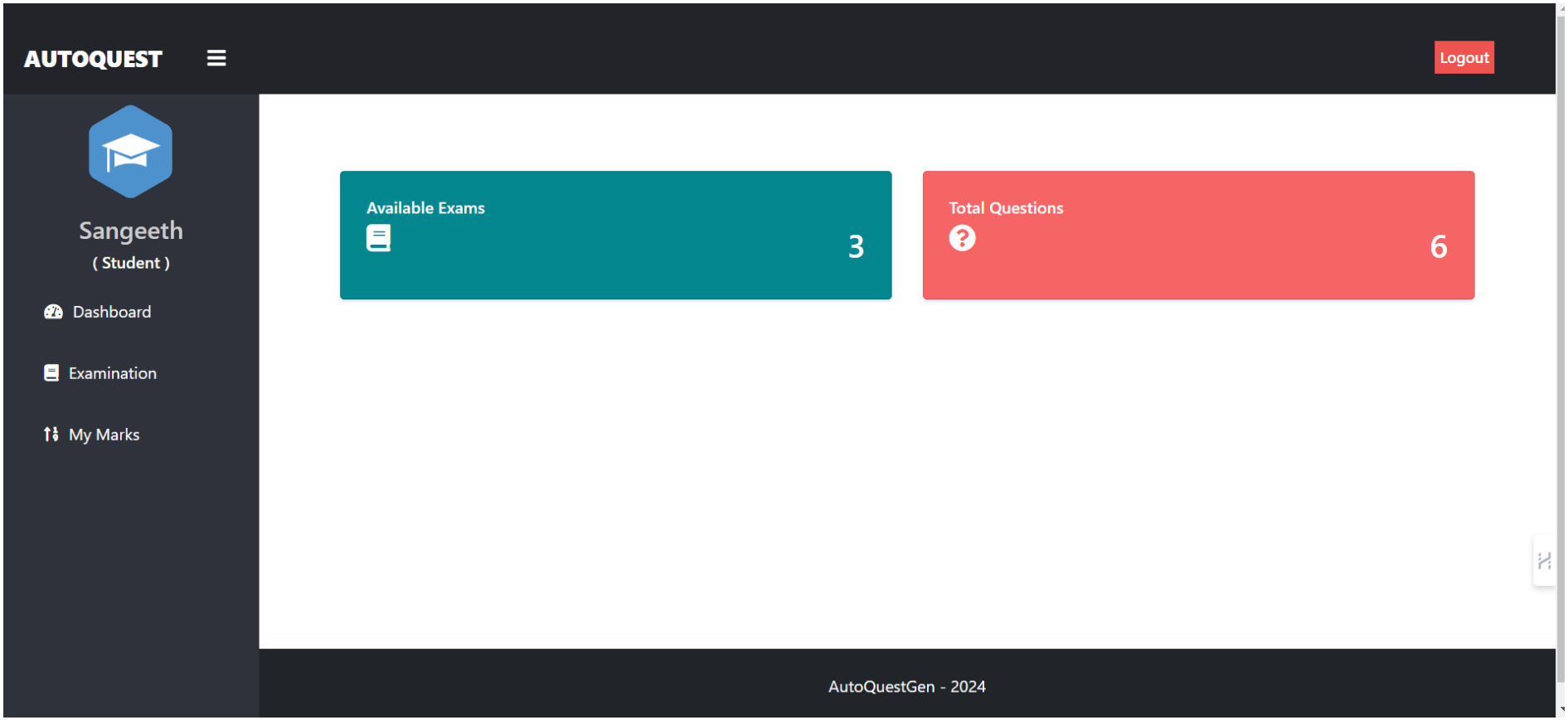

Figure 15: Student Dashboard

View Available Exams

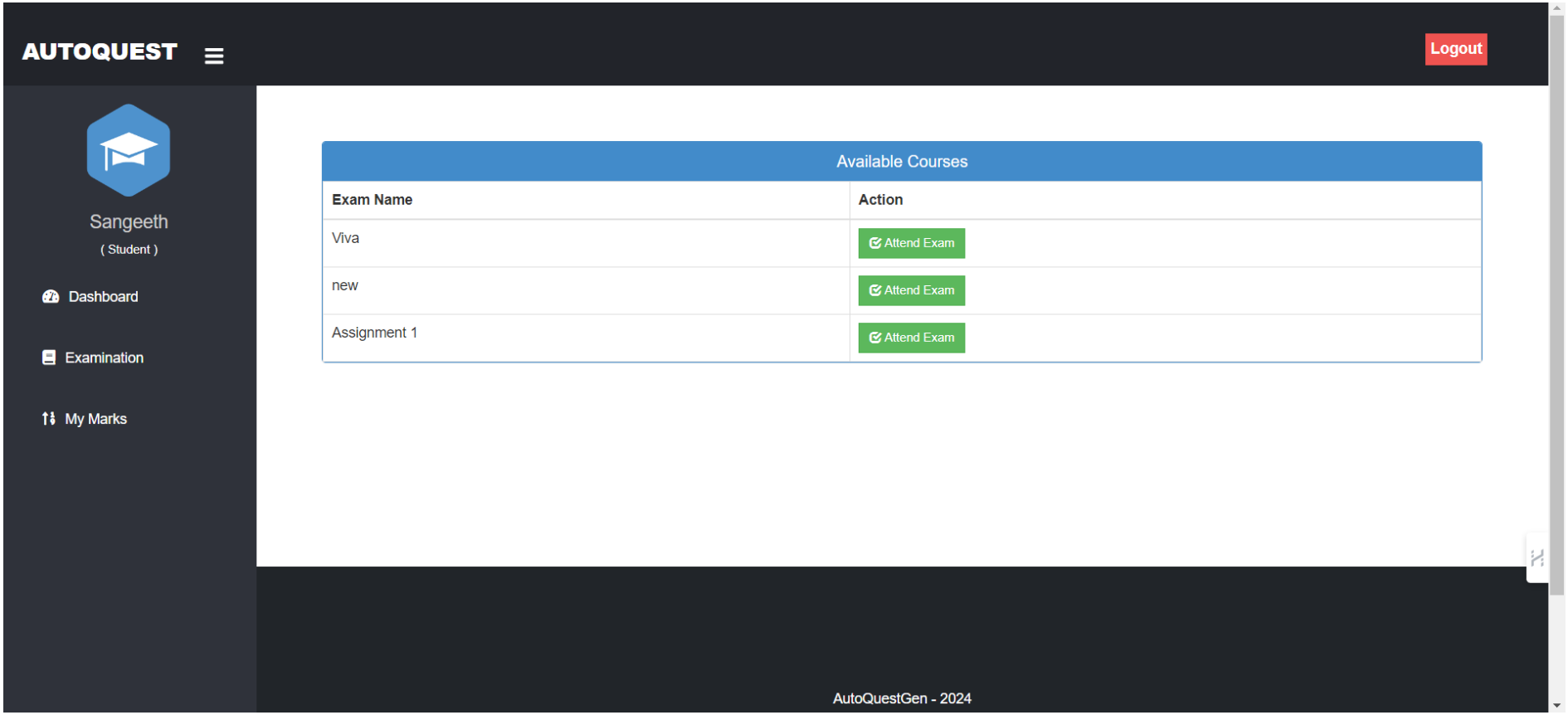

Figure 16: View Available Exams

Exam Details

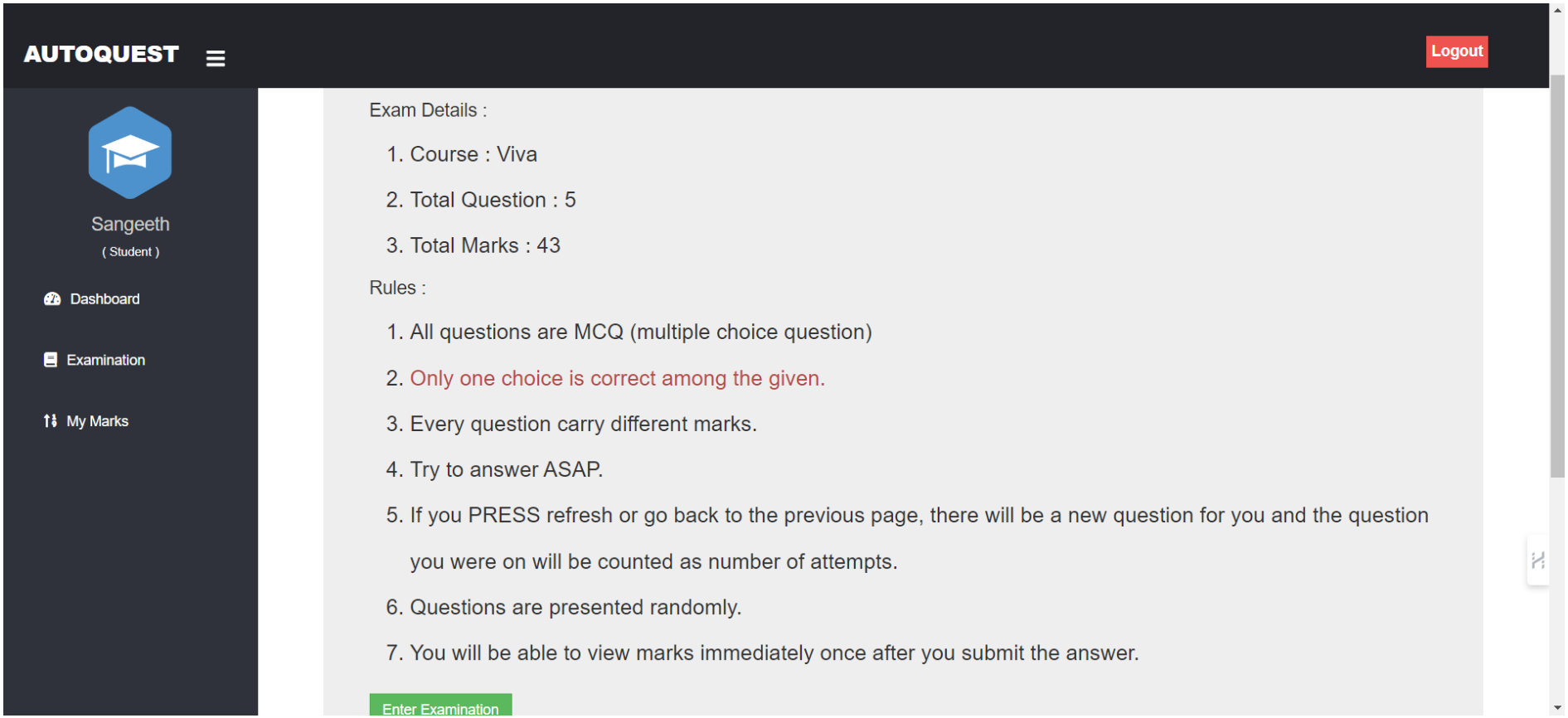

Figure 17: Exam Details

Attend Exam

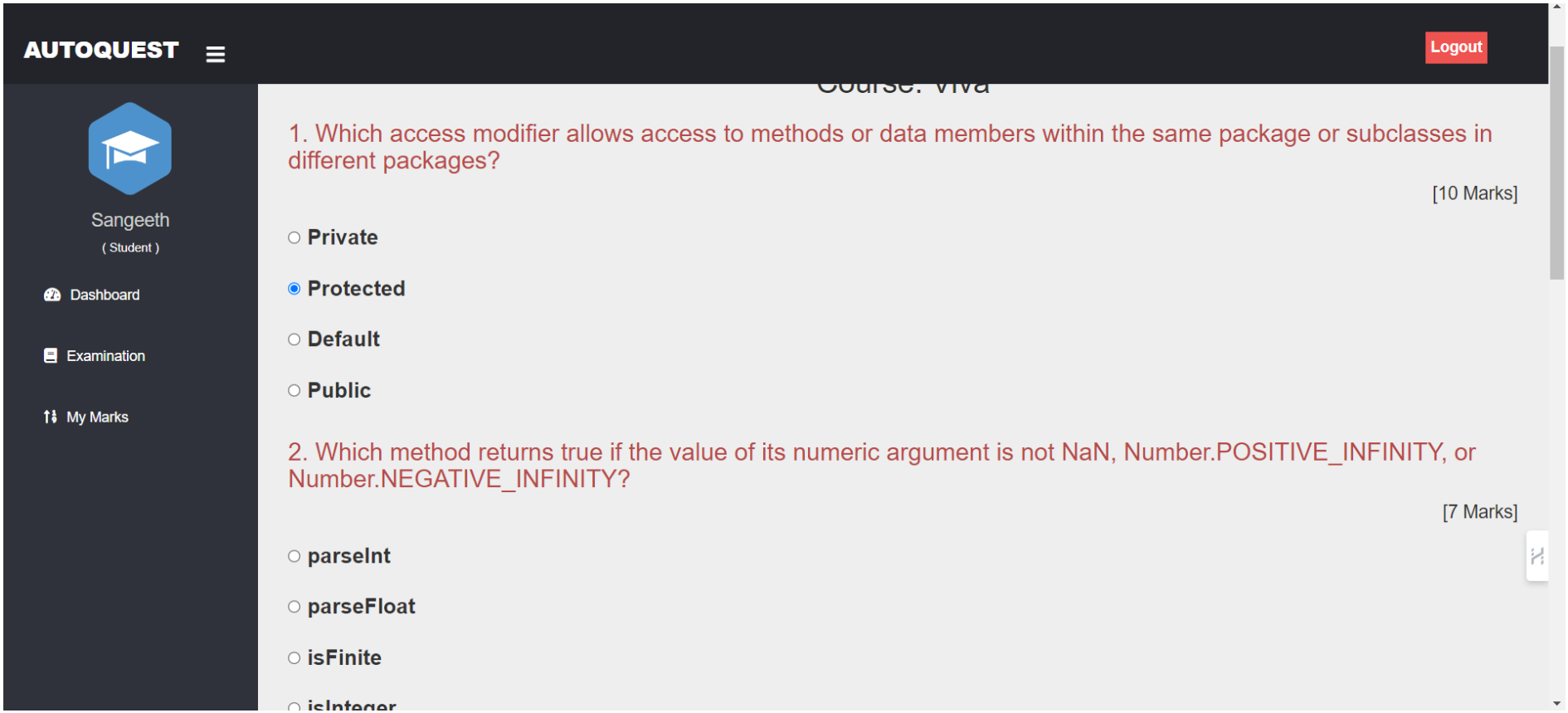

Figure 18: Attend Exam

View Marks

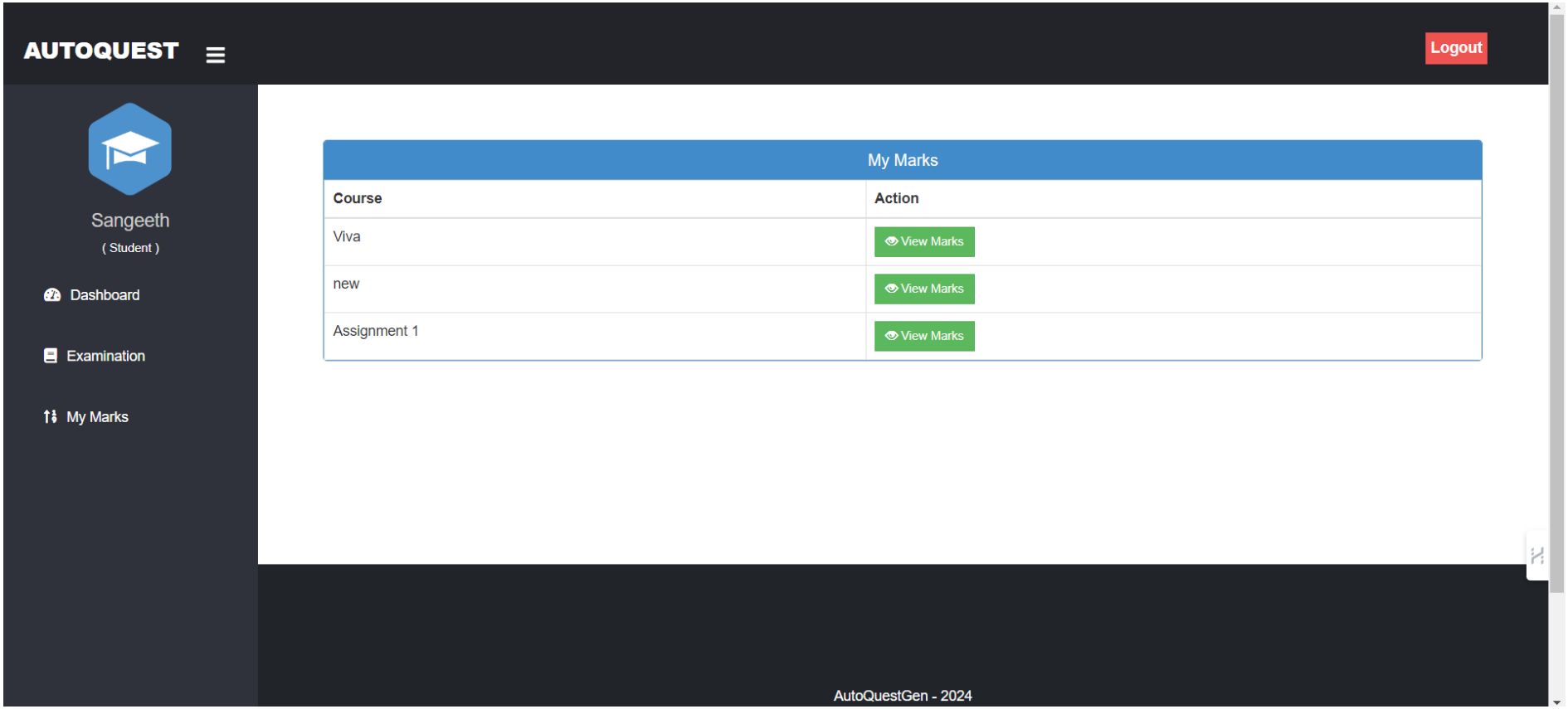

Figure 19: View Marks

View Individual Marks

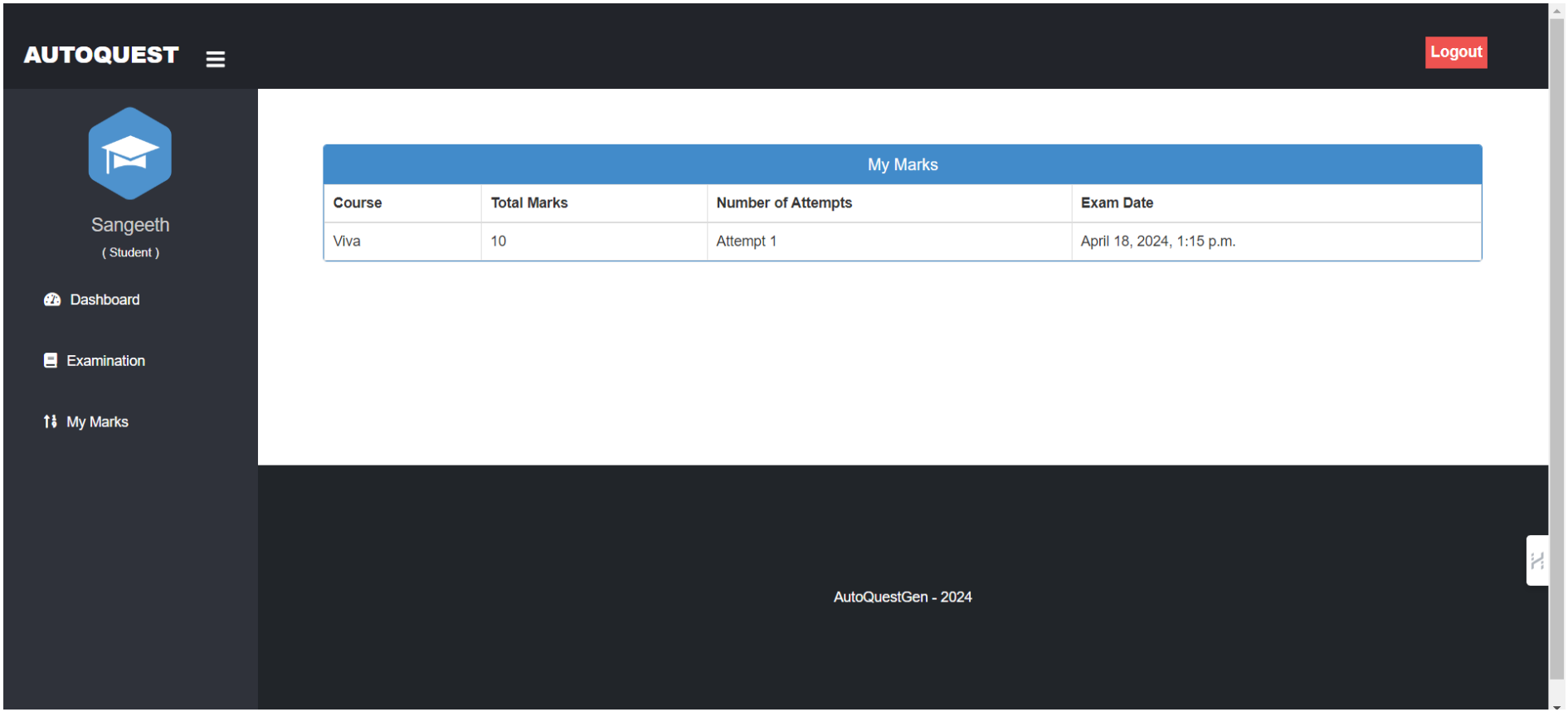

Figure 20: View Individual Marks

Admin Dashboard

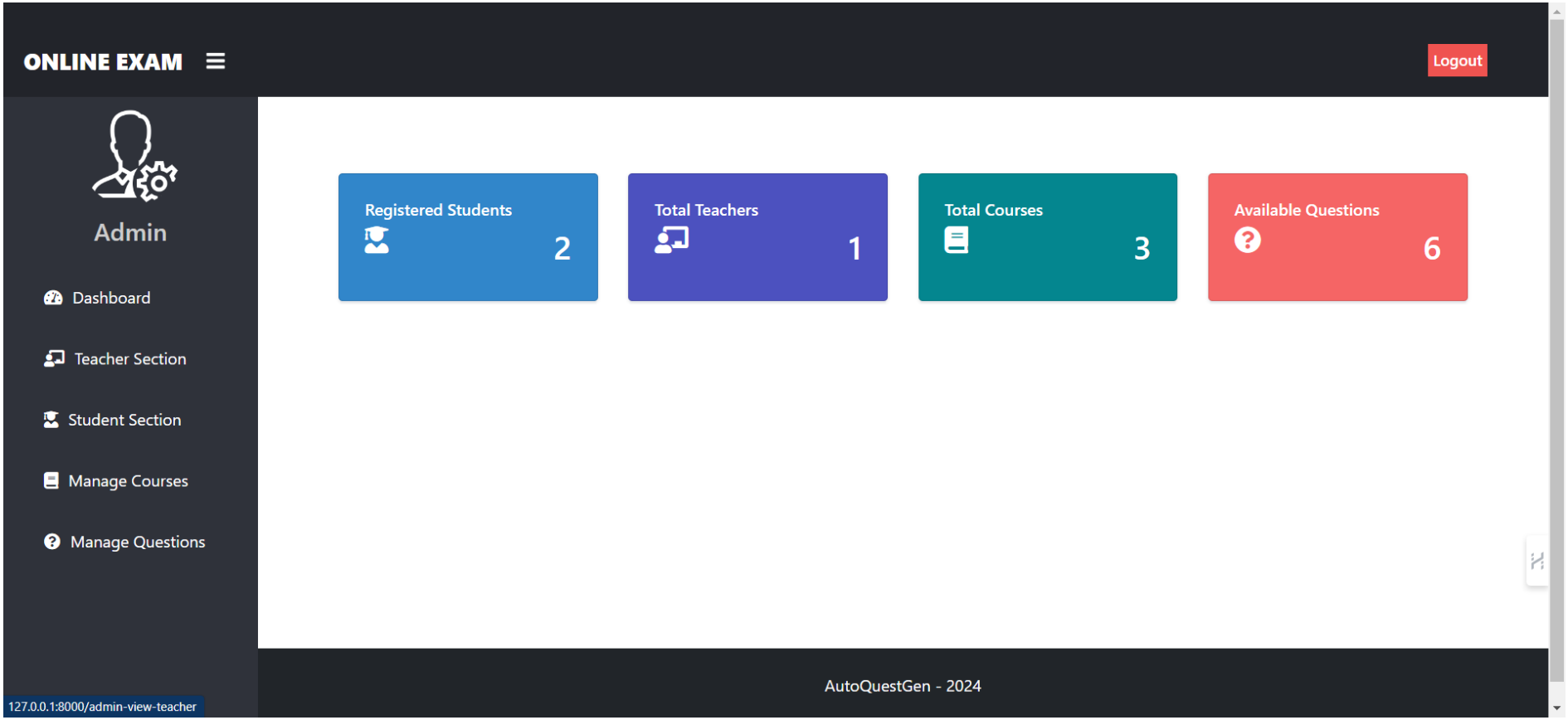

Figure 21: Admin Dashboard

Teacher Section

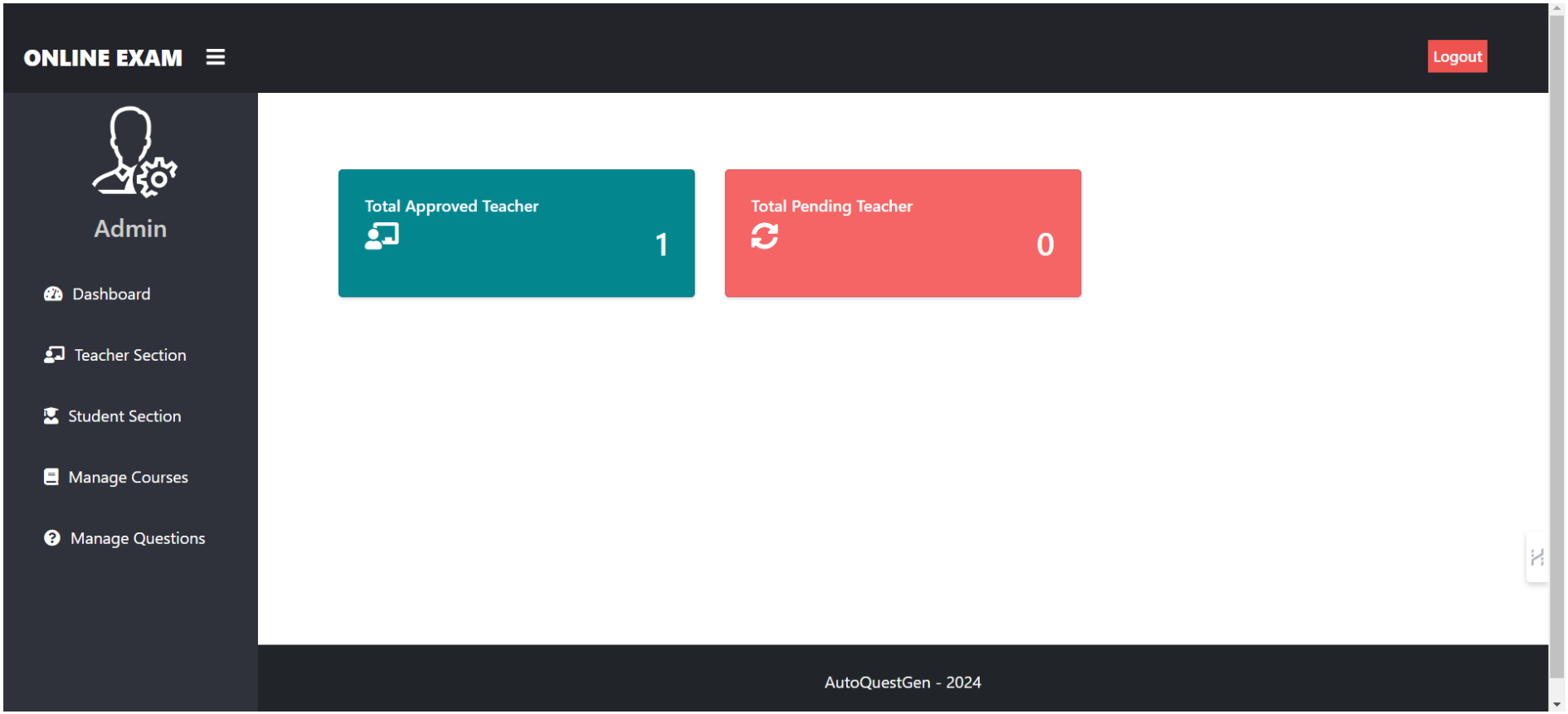

Figure 22: Teaher Section

Approved Teacher

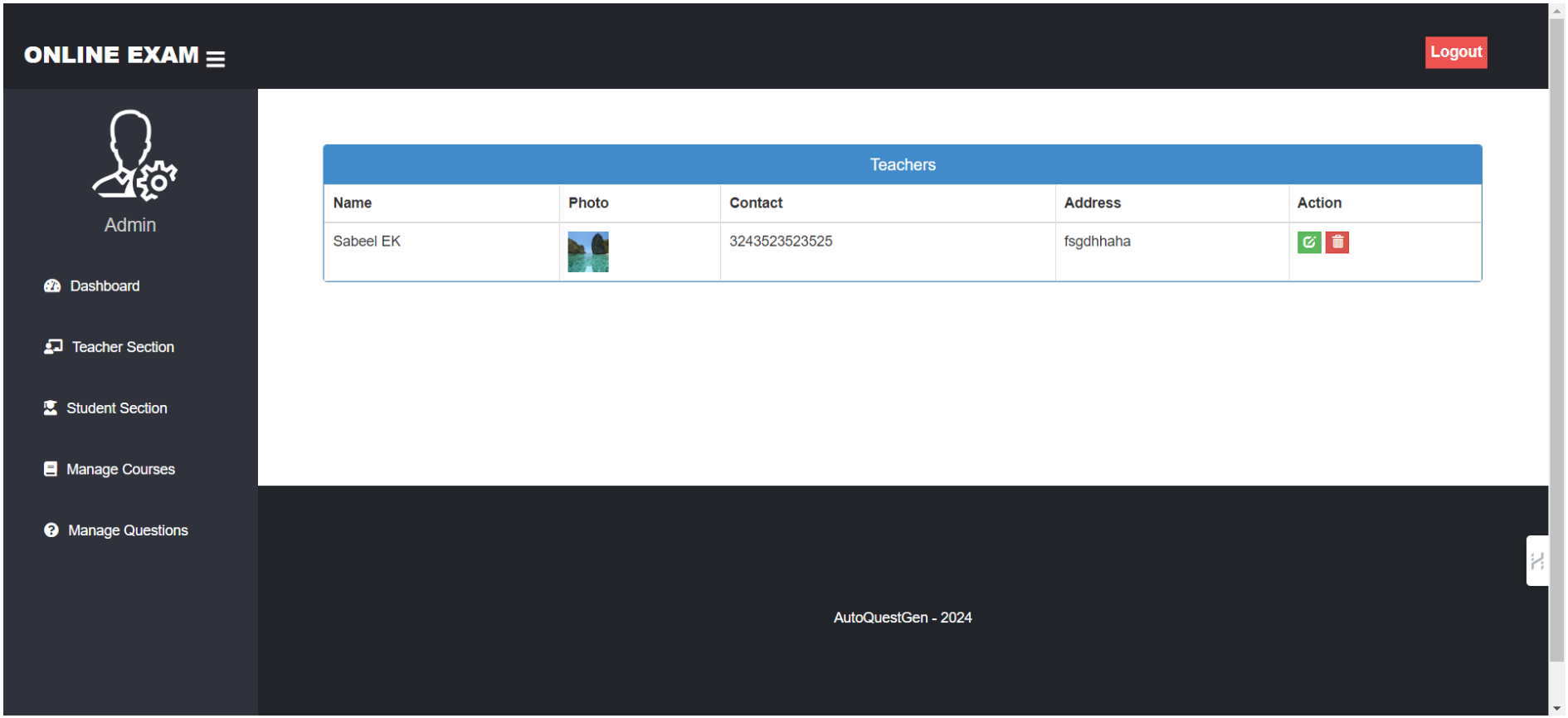

Figure 23: Enter Caption

Approval Pending Teachers

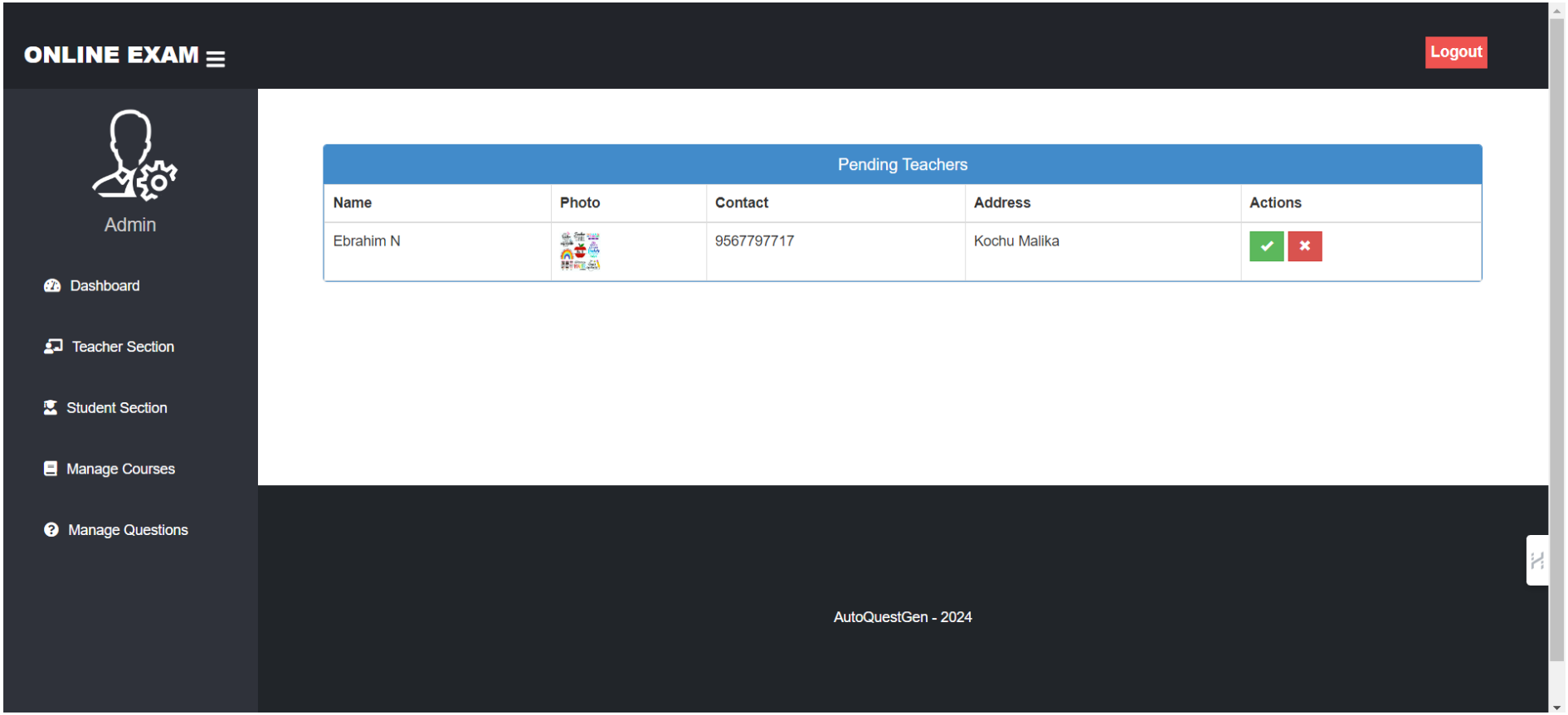

Figure 24: Approval Pending Teachers

Student Section

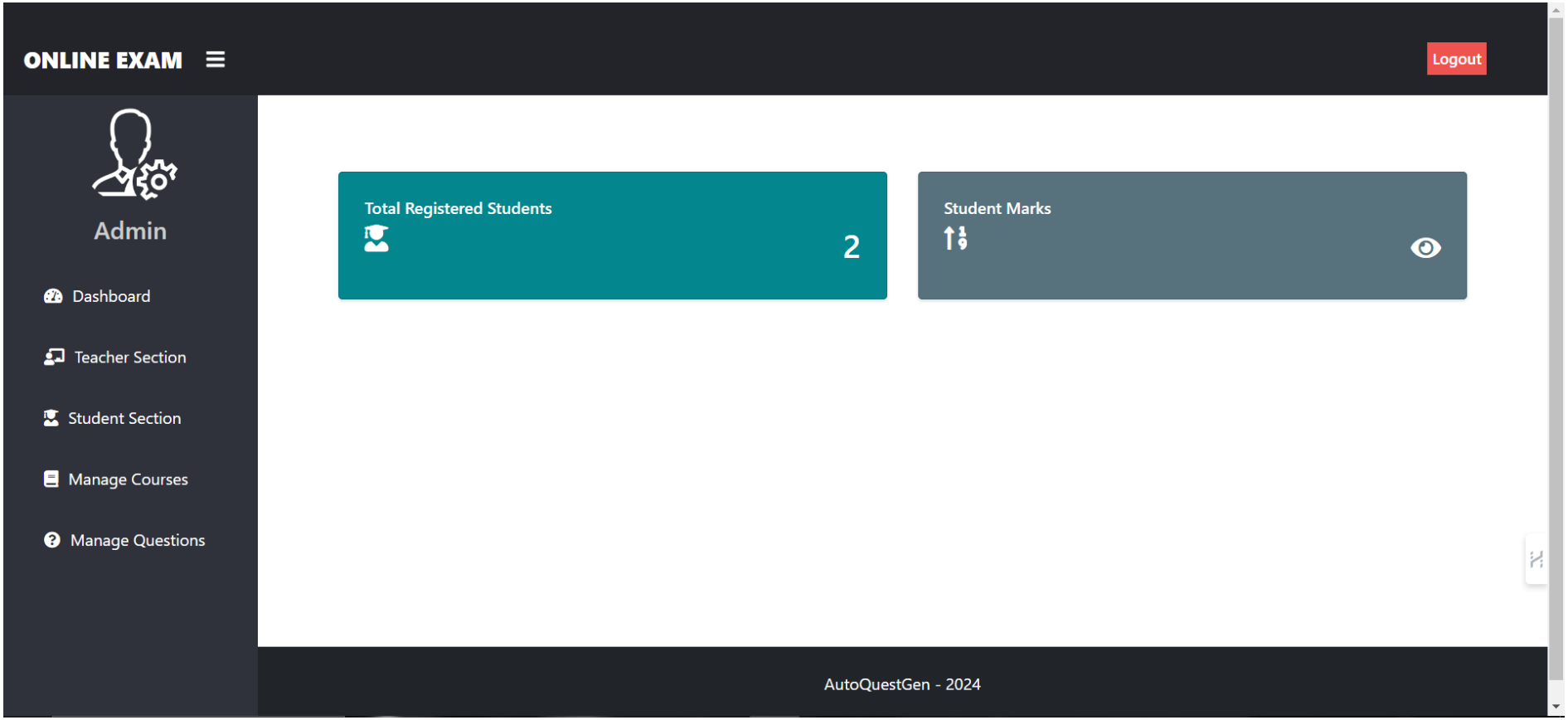

Figure 25: Student Section

View Students

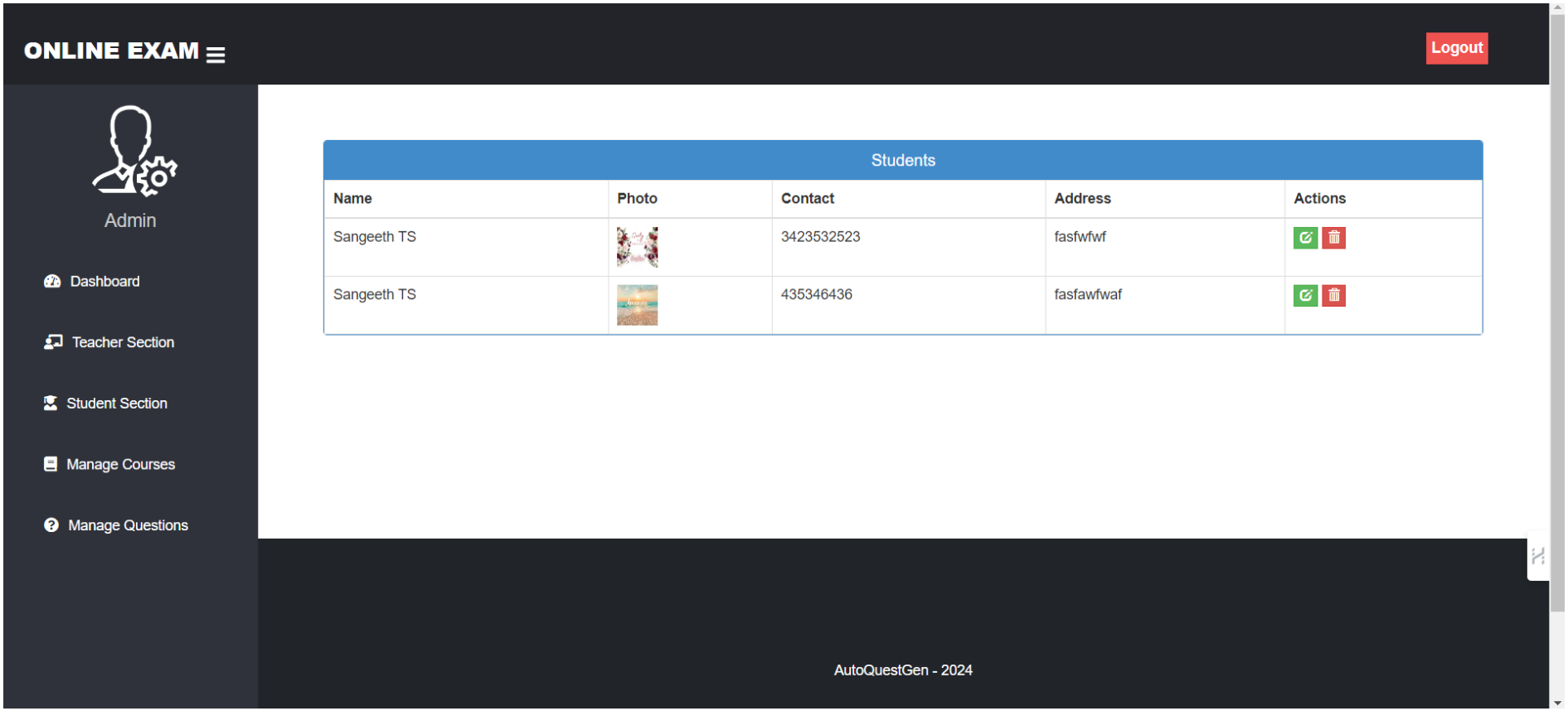

Figure 26: View Students

Manage Exams

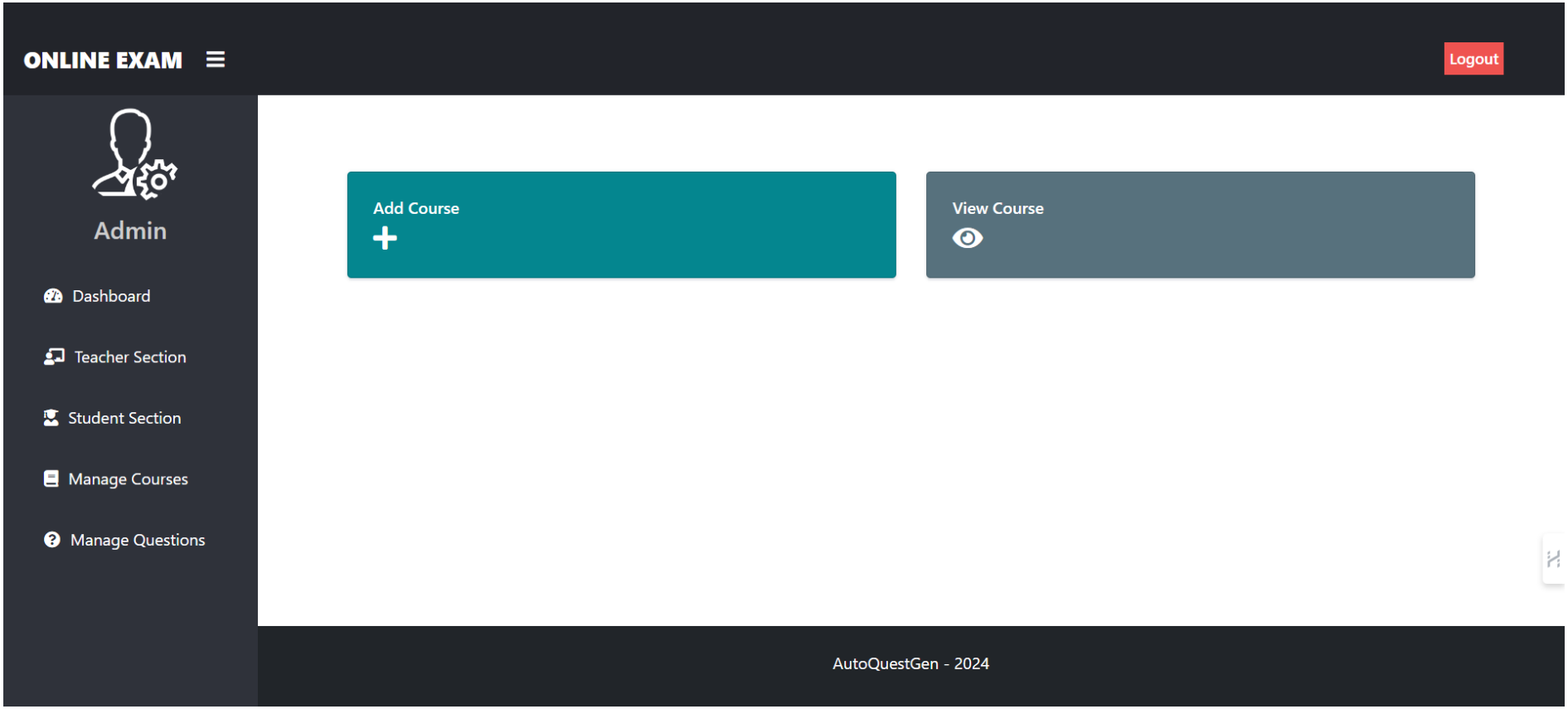

Figure 27: Manage Exams

View Exams

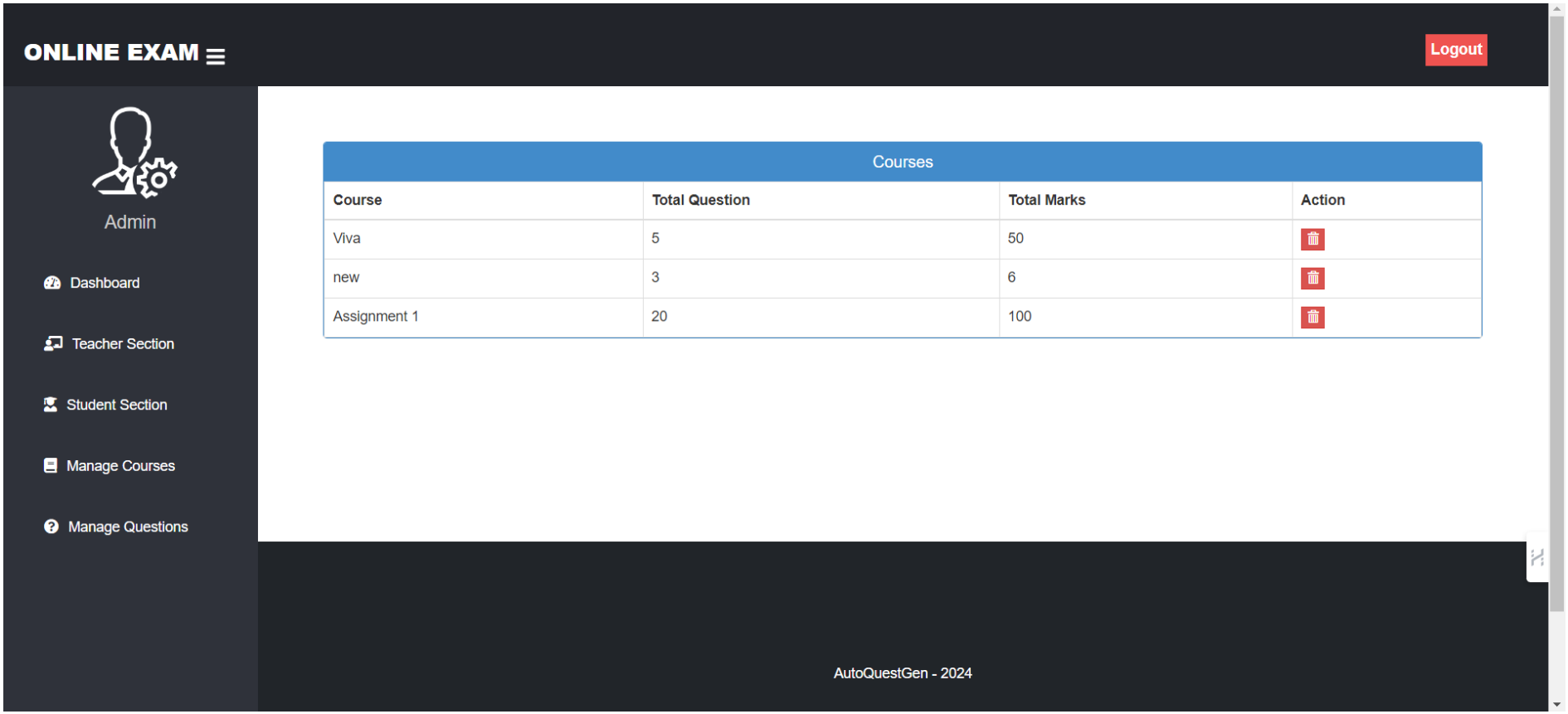

Figure 28: View Exams

Manage Questions

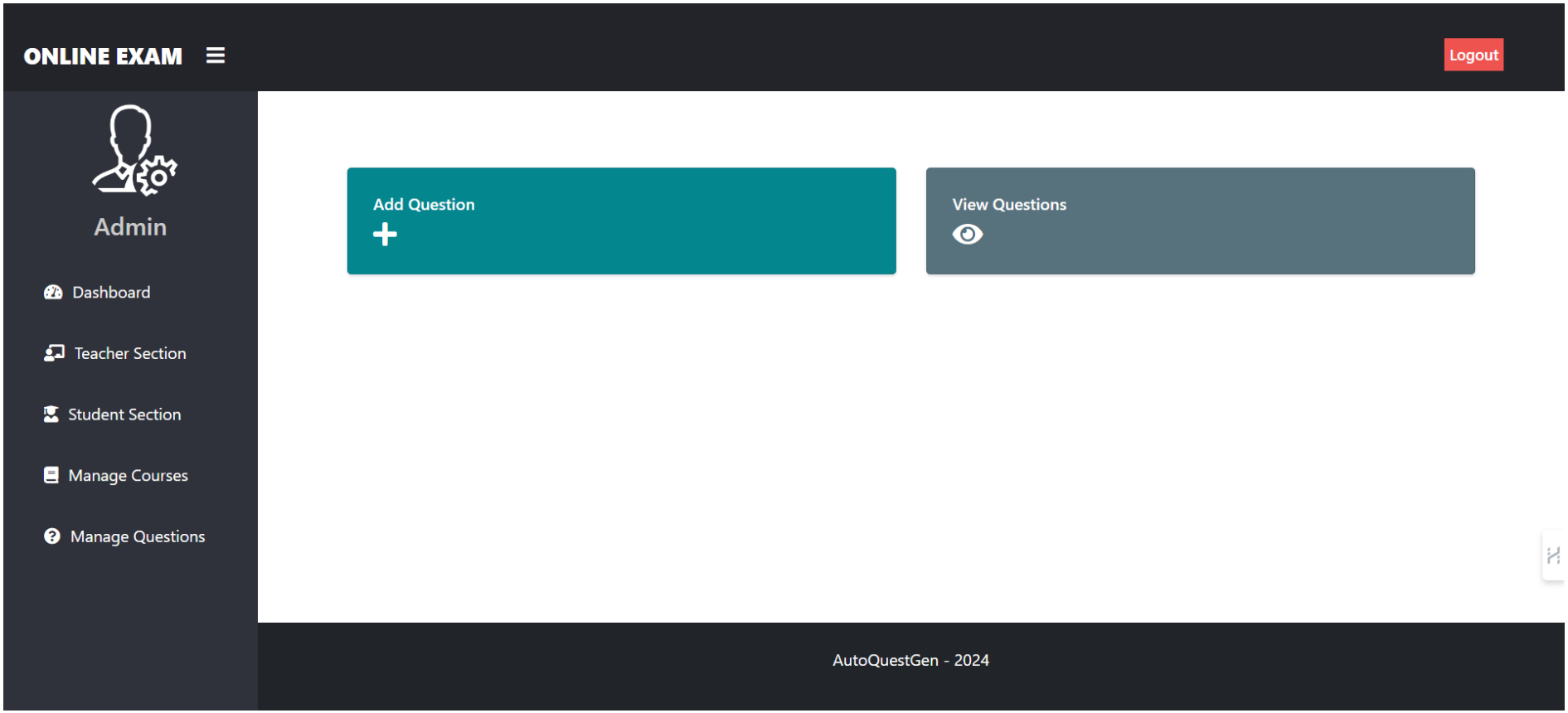

Figure 29: Manage Questions

Conclusion

- Autoquestgen can help teachers simplify the process of conducting exams.
- It can help eliminate the need for evaluation of question papers.
- Students can get their marks instantly.

Future Scope

- Implementation of Subjective Question and its evaluation
- Inputs in multiple formats like Pdf and Websites

References

We have taken reference from the following:

1. Surbhi Choudhary, Abdul Rais Abdul Waheed, Shrutika Gawandi and Kavita Joshi, âQuestion Paper Generator System,â International Journal of Computer Science Trends and Technology, vol. 3, issue 5, Sept â Oct 2015.
2. Prita Patil and Kavita Shirsat, âAn Integrated Automated Paperless Academic Module for Education Institutes,â International Journal of Engineering Science Invention Research and Development, vol. I, issue IX, March 2015.
3. Mrunal Patangare, Rushikesh Pangare, Shreyas Dorle, Uday Biradar, Kaustubh Kale, âAndroid Based Exam Paper Generatorâ Proceeding of the Second International Conference on Inventive Systems and Control (ICISC 2018).
4. Suraj Kamya, Madhuri Sachdeva, Navdeep Dhaliwal, Sonit Singh, "Fuzzy Logic Based Intelligent Question Paper Generator", vol. I, April 2014
5. Gauri Nalawade, Rekha Ramesh, "Automatic Generation of

References

6. G. Jagadamba, Chaya Shree G, "Online Subjective Answer Verifying System Using Artificial Intelligence", vol. II, May 2017
7. Selvia Ferdiana Kusuma, Daniel Oranova Siahaan, Chastine Fatichah, Mohammad Farid Naufal, "Automatic Question Generation With Classification Based On Mind Map", June 2018
8. Suraj Kamya, Madhuri Sachdeva, Navdeep Dhaliwal, Sonit Singh, "Hand-Written and Machine-Printed Text Classification", 2018
9. Muhammad Farrukh Bashir, Hamza Arshad, Abdul Rehman Javed, Natalia Kryvinska, Shahab S. Band, "Subjective Answers Evaluation", April 2014
10. Rusyaizila Ramli, Salvester Sivan Jr, Husniza Razalli, "A Review on Automated Examination Question Paper Template Generator", May 2020

Thank You