Mobile App For MCQ Exam Management and Result Generation

By

B.M. Rahat Almas
203-15-3914

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Ms. Amatul Bushra Akhi
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Mr. Amir Sohel
Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

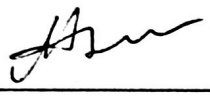
DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

January 13, 2025

APPROVAL

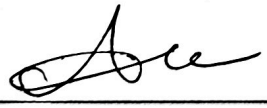
This Project titled Mobile App For **MCQ Exam Management and Result Generation** submitted by B.M. Rahat Almas to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **13-01-2025.**

BOARD OF EXAMINERS

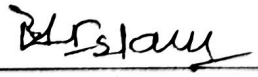


Ms. Nazmun Nessa Moon (NNM)
Board Chairman
Associate Professor, Department of CSE, FSIT
Daffodil International University

Dewan Mamun Raza (DMR)
Internal Examiner 1
Assistant Professor, Department of CSE, FSIT
Daffodil International University



Mr. Abdullah Al Mamun (AAM)
Internal Examiner 2
Lecturer, Department of CSE, FSIT
Daffodil International University



Dr. Md. Manowarul Islam
External Examiner
Associate Professor, Department of CSE
Jagannath University

©Daffodil International University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Ms. Amatul Bushra Akhi, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

13.01.25

Ms. Amatul Bushra Akhi
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Mr. Amir Sohel
Senior Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Rahat Almas

B.M. Rahat Almas
203-15-3914
Department of Computer Science and Engineering
Daffodil International University

©Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Ms. Amatul Bushra Akhi**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Image Processing** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

This project presents an MCQ Exam Management System that simplifies and automates the multiple choice questions exam management, grading, and result generation process. An administrator can manage various aspects of the exam process, including adding questions, assigning applicants to tests, generating new exams, and adding scholar data. A unique feature of this system is the integration of Optical Mark Recognition (OMR) technology, which allows the administrator to upload completed OMR sheets filled out by candidates. One of the most useful features of this system is the integration of Optical Mark Recognition (OMR) technology. This feature allows administrators to upload OMR sheets completed by students. The system reads the answers on these sheets, grades them automatically, and generates individual results for each student. These results are securely stored in a centralized database, making them easy to access and manage. To make things more convenient, the system provides an option to download exam results in PDF format. This helps administrators and students view and share results easily without the need for printed copies. By automating tasks like grading and result generation, the system saves time, reduces errors, and makes exam management more efficient. This MCQ Exam Management System is a user-friendly tool that simplifies the exam process. It replaces manual grading and paperwork, making exams faster, more accurate, and easier for everyone involved. With the help of this solution, MCQ tests may be handled more effectively, accurately, and user-friendly, doing away with the need for manual grading and paper-based result distribution.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In today's fast-paced and technology-driven world, the need for efficient and automated systems in education has become increasingly important. Multiple Choice Questions (MCQ) exams are a widely adopted assessment method due to their ability to evaluate large groups of candidates efficiently and objectively. However, traditional methods of managing such exams, including question preparation, manual grading, and result distribution, often lead to inefficiencies, increased workload, and the risk of human error.
The MCQ Exam Management System aims to address these challenges by providing an integrated platform that simplifies and automates various aspects of the exam process. The system is designed to help administrators manage tasks such as creating and assigning exams, storing and updating student information, and processing results. One of the standout features of this solution is the incorporation of Optical Mark Recognition (OMR) technology, which streamlines the evaluation process by extracting answers directly from OMR sheets filled out by candidates. This system not only improves the accuracy and efficiency of the grading but also ensures secure storage and easy access to the exam results via a centralized database. In addition, the option to download results in PDF format makes it convenient for administrators to share results with candidates and other stakeholders. Using automation and digital solutions, the MCQ Exam Management System revolutionizes the traditional exam workflow, offering a user-friendly and robust approach to modern educational assessments.

1.2 Motivation

Traditional methods of organizing MCQ exams are often time-consuming, prone to human errors during grading, and require significant manual effort for the distribution of results. These inefficiencies not only increase the workload for educators and administrators, but also delay the overall assessment process, impacting the timely delivery of feedback to students. As educational institutions strive to adopt more streamlined and automated

solutions, there is a pressing need for systems that can minimize errors, reduce manual result generation time, and accelerate the examination workflow. The integration of technology into exam management processes, such as automated grading using Optical Mark Recognition (OMR) and centralized data storage, offers a promising solution to these challenges. This project is motivated by the desire to bridge the gap between traditional exam practices Developingern technological capabilities, ensuring precision, efficiency, and ease of access for all involved users.

1.3 Objectives

The primary objectives of MCQ Exam Management System are:

- Develop a platform for Creating exam, question papers and storing student information.
- Dedicating server to implement and serve optical mark recognition (OMR) technology to that will ensure accurate evaluation of answer sheets.
- Designing centralized database to store and manage all the exam and students record.
- Features and functionality for the administrators to create and download PDFs of created questions, OMR sheets and generated results.
- Easy to use user interface design to access all the features comfortably.
- If workloads grows server will able to rescale to match the loads.
- Ensured Security to protect sensitive student and exam data from unauthorized access.

1.4 Methodology

The methodology section explains the approach you'll take to develop the MCQ Exam Management System. Here's a sample outline:

a. Requirement Analysis:

- Identify and gather requirements from stakeholders such as educators, administrators, and students.
- Define the key features, functionalities, and constraints of the system.

b. System Design:

- Develop an architecture for the system, incorporating modules for exam creation, OMR processing, result generation, and database management.
- Use flowcharts and data models to map out the system's operations.

c. Development:

- Use dart and javascript programming language to implement the features. Frameworks (expressjs, flutter).
- Integrate Optical Mark Recognition (OMR) technology for automated answer sheet evaluation.
- Develop a user-friendly web interface with responsive design.

d. Database Setup:

- Design a centralized relational database using MySQL to store student information, question banks, and exam results.

e. Testing and Debugging:

- Conduct unit, integration and system testing to ensure that the platform is functional, accurate, and secure.
- Test OMR accuracy with sample answer sheets.

f. Deployment:

- Host the system on a scalable server infrastructure to handle varying workloads.

g. Evaluation and Feedback:

- Collect feedback from users to refine the system and address any usability or performance problems.

1.5 Project Outcome

- A fully functional MCQ Exam Management System with features for exam creation, student record management, and automated grading using OMR technology.
- A centralized database that securely stores all exam-related data and allows easy retrieval for reporting and analysis.
- Improved accuracy and speed in exam grading, reducing human error and administrative workload.
- A user-friendly interface that enables administrators to efficiently manage the entire exam lifecycle.
- The ability to generate and download exam results and OMR sheets in PDF format, enhancing accessibility and sharing.
- Scalability and reliability of the system to handle large numbers of users and exams simultaneously.
- Enhanced security features to protect sensitive data and ensure privacy compliance.

1.6 Organization of the Report

This section describes the remaining structure of the entire document.

Table 1.1: Organization of the report

Chapters	Overview
Chapter 2	Compares among the existing systems and the designed model.
Chapter 3	Details of the step by step approach used to design & develop the system
Chapter 4	Explains of the system design, implementation and architecture of the system.
Chapter 5	Details of the testing and results
Chapter 6	Highlights the significance of the system and conclusion.

Chapter 2

Background

2.1 Introduction

In the digital age, the automation of educational assessments has become increasingly important to improve efficiency, accuracy, and accessibility. Multiple Choice Questions (MCQ) exams are a widely used method for evaluating large groups of candidates due to their objective nature and ease of grading. Various applications have been developed to streamline this process, integrating Optical Mark Recognition (OMR) technology to automate answer sheet evaluation. These applications offer a range of features tailored to different user needs, from exam creation and grading to performance tracking and report generation.

2.2 Literature Review

2.2.1 Similar Applications

There are many applications available regarding MCQs and OMR sheet scanners on the internet. Each application has their own unique features, usages, and targeted users.

EvalBee[1] is an OMR sheet scanner application which was developed by Ekodriod labs. In this application, admin can create new exams and upload and update questions. Through this application users can generate OMR sheets as well as scan the paper OMR sheets. This app also includes features such as analysis reports, download the sheets, and make answer keys.

Dynamic OMR sheet[2], was developed by Reak AbC & Co, for UPSC, JEE, NEET, BPSE and all other related examinations. This application includes features such as creating OMR sheets, setting timer, and generating reports. Admins can input correct answers into the system and upload students' answer sheets. This app scans their sheets and performs an automatic analysis of the responses. After analyzing, this app will generate reports. Users can also track their previous examinations and monitor progress.

Live MCQ[3] was developed by the CrackTech company. It is specially designed for the job seeking candidates. The app focuses on only the job market including BCS(Bangladesh

2.3 Gap Analysis

Table 2.1: Gap Analysis

Features	Eval Bee	Dynamic OMR	Live MCQ	Aspose. OMR	Proposed system
Create Exam	Yes	No	No	No	Yes
Candidate Register	Yes	Yes	No	No	Yes
Add scholar	No	No	No	No	Yes
OMR pdf	No	Yes	No	Yes	Yes
Result PDF	Yes	No	No	Yes	Yes
Offline exam data management	No	No	Yes	Yes	Yes
Admin control	Yes	Yes	Yes	Yes	Yes
Public user	Yes	Yes	Yes	Yes	No
OMR data extraction	No	Yes	No	No	Yes

Civil Service), banking, and teaching. This is an interactive quiz system application where candidates can not only participate in the quizzes but also can access the explanation of the correct answers. This app features graphical representations of top-ranked candidates and a leaderboard to foster competitive engagement.

Aspose.OMR[4] developed by Aspose Cloud, is an application designed for creating and recognizing answer sheets with smartphones. Administrators can input correct answers into the system and scan students' OMR sheets and the app scans the sheets and identifies if the answer is correct or not.

Britto[5] is a learning and testing application by Innospace Infotech Ltd targeting HSC and SSC candidates. It is an offline application where students can evaluate themselves by taking the quizzes, making it a valuable tool for academic preparation.

OMR[6], developed by Surya Vedantam, is also a typical quiz based application utilizing OMR sheet technology. Admin can create quizzes, set timer, and enter correct answers into the system. Students can view the questions and input their answers. This application will automatically analyze the answers and generate results.

MCQS[7] for SSC, HSC, medical, computing exam candidates. This app requires no internet to run. Users can select the exam they wish to participate in, take quizzes, and assess their performance without requiring an internet connection.

2.4 Summary

The market offers several applications for automating MCQ based assessments and using OMR technology to simplify grading and result generation. There are several apps in the market such as EvalBee which focus on exam creation, automated grading, and progress tracking, while Live MCQ targets job-seekers with interactive quizzes and leaderboards. Others, like Aspose.OMR and Britto, cater to mobile users and offline self-assessments, respectively. However, these tools often target specific audiences or functionalities. In

contrast, MCQ Exam Management System offers a comprehensive solution, integrating features like exam creation, secure centralized result storage, automated OMR based grading, and scalability. Unlike the limited scope of existing tools, this system caters to diverse users and ensures a robust, user-friendly platform, making it an efficient solution for modern educational needs.

Chapter 3

Research Methodology

3.1 Methodology

3.1.1 Overview

The MCQ Exam Management System is designed with a client-server architecture, integrating an application based platform for administrators and students. The system includes a backend that handles exam creation, grading, and database management. A key feature is the integration of Optical Mark Recognition (OMR) technology for automated grading, which scans and processes OMR sheets to generate results. In this section, the diagrams including Usercase, Entity- Relation(ER) diagram, Unified Modeling Laguage(UML) diagram are drawn. The visual representation of the designed application is illustrated here. Screenshots of the application are provided here as User Interface(UI) in the UI design subsection.

3.1.2 Proposed Methodology

During planning and designing this application, I have followed SDLC model [fig 3.1]. The Software Development Life Cycle (SDLC) is a foundational concept in software en-

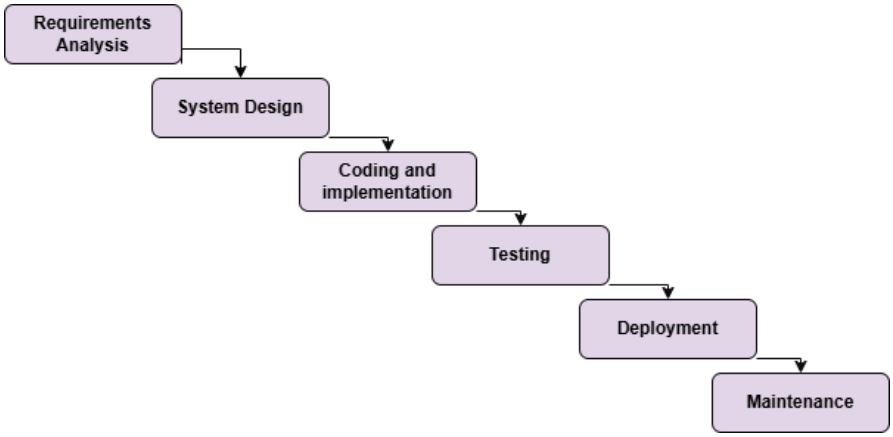


Figure 3.1: SDLC model

gineering. It provides a systematic framework for planning, developing, and maintaining software. It ensures the end product meets or exceeds customer expectations. It maintains timelines for project milestones and final release. The SDLC model steps are described below:

i. Requirements Analysis:

- Gathering functional and non-functional requirements from stakeholders.
- Identifying the challenges in existing systems, such as manual grading inefficiencies and lack of scalability.
- This also identifies the functional and non-functional requirements such as security, performance, usability.

ii. System Design:

- Designing a backend architecture to support OMR integration, grading automation, and centralized database management.
- Developing a frontend interface to enable seamless interactions for both administrators and students.

iii. Implementation:

- Express.js for backend development and Flutter for creating a responsive frontend.
- Integrating libraries for OMR processing and secure database handling.
- Development of features and functionalities. Adherence to coding standards and guidelines.
- The version control is done using tools like Git.

iv. Testing:

- Conducting unit tests for individual components, integration tests for combined modules, and user acceptance testing to ensure reliability and usability.

v. Deployment:

- Hosting the system on a secure server with robust infrastructure to handle scaling requirements.

vi. Maintenance:

- Regular updates to improve performance, address bugs, and incorporate new features.

3.1.3 Functional & Non Functional requirements

Functional requirements

Functional requirements describe what the system should do. These are directly related to the specific functionalities or services a system must provide to fulfill its purpose. They define behavior and features of the system.

- **Exam Management:** Admin can create, update, and delete exams. They can assign exams to specific groups or candidates.
- **Student Data Management:** Admin can add, update, and delete student records.
- **Question Bank Management:** Admin can upload, categorize, and modify questions.
- **OMR Integration:** Admin can generate OMR sheets for candidates as well as upload scanned OMR sheets for automated answer extraction.
- **Automated Grading:** The system extracts answers from OMR sheets and matches them against the answer key and calculates scores.
- **Result Management:** Generates individual and consolidated exam results.
- **Centralized Database:** Securely store student data, question banks, exam details, and results.
- **User Authentication and Authorization:** Secure login system for admins.

Non-Functional Requirements

Non-functional requirements define the quality attributes, system constraints, and operational standards. They focus on how the system performs the defined functions.

- **Performance:** The system processes OMR sheets and generates results.
- **Scalability:** The system is capable of handling increasing numbers of users, exams, and data without requiring significant redesign.
- **Reliability:** This app ensures 99.9% system uptime for uninterrupted usage.
- **Usability:** This app provides an intuitive and user-friendly management. administrators.
- **Security:** Encrypt sensitive data, including login credentials and exam results. Implement protection against SQL injection, XSS, and other common vulnerabilities.

3.1.4 Context Diagram

ER Diagram

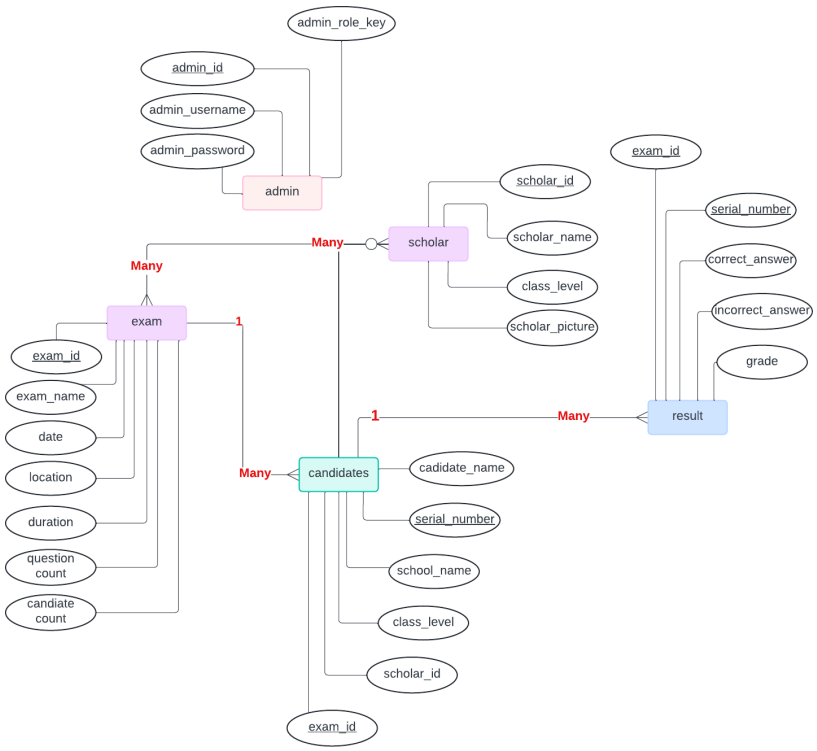


Figure 3.2: ER diagram

UML Diagram

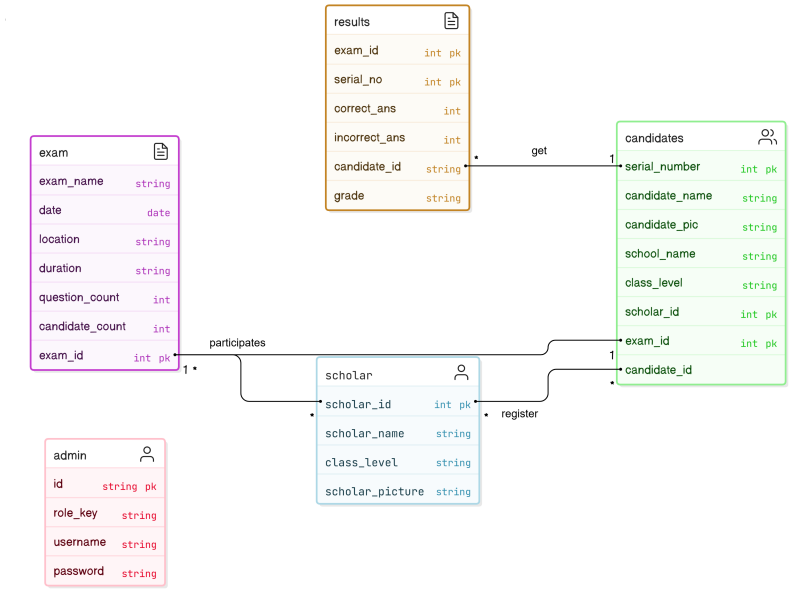


Figure 3.3: UML Diagram

User Case Diagram

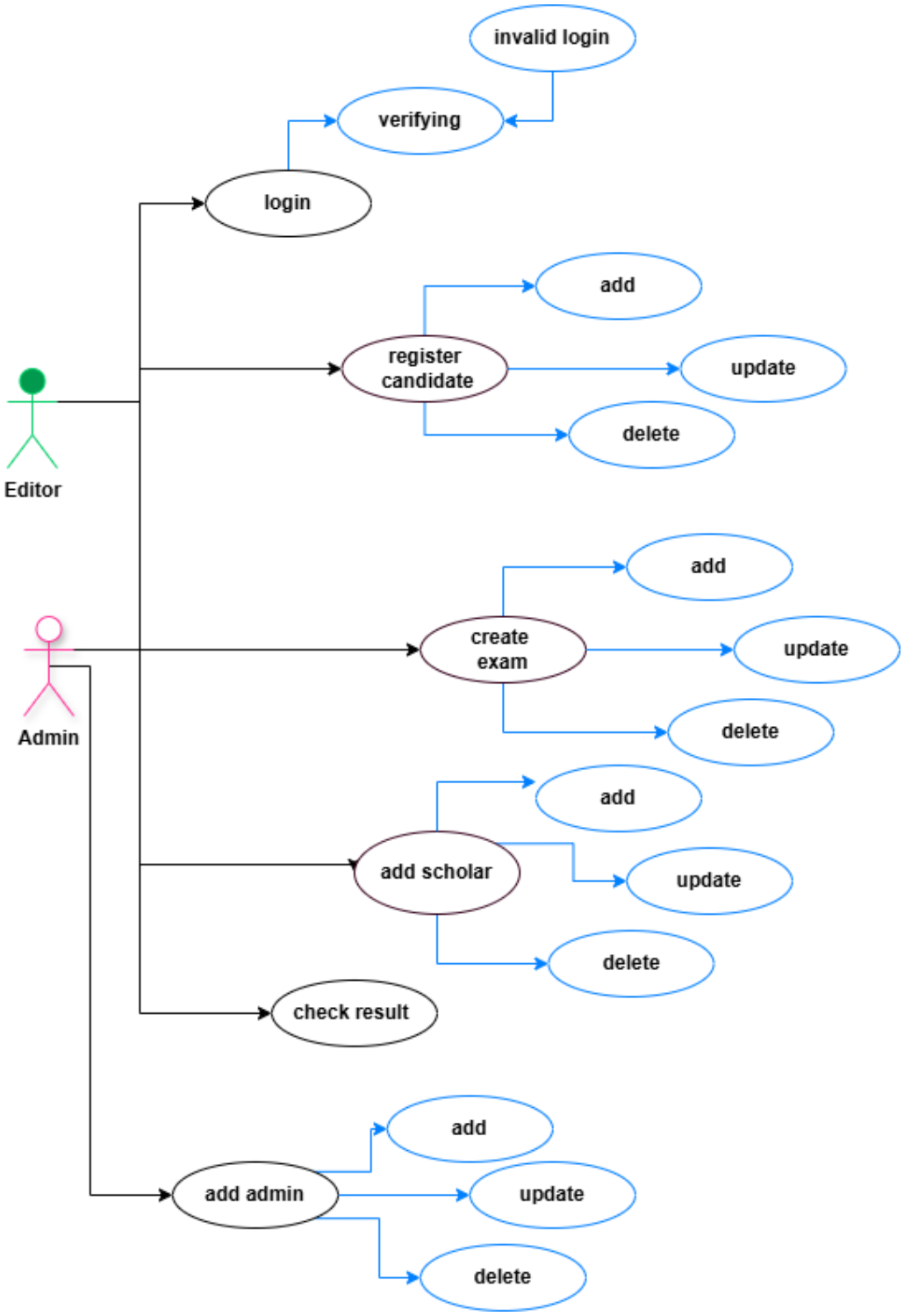


Figure 3.4: User Case diagram

3.1.5 UI Design

Front End Design

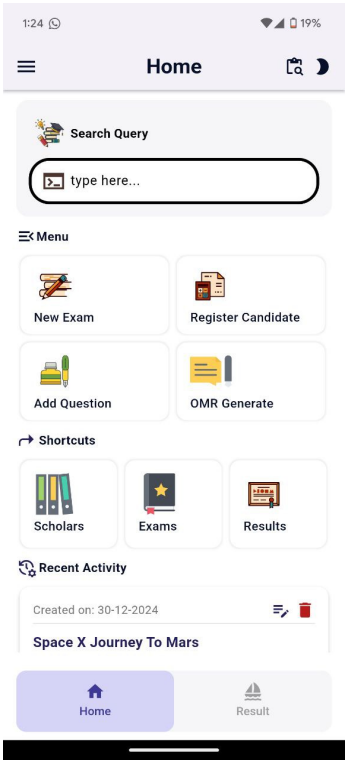


Figure 3.5: Home Page

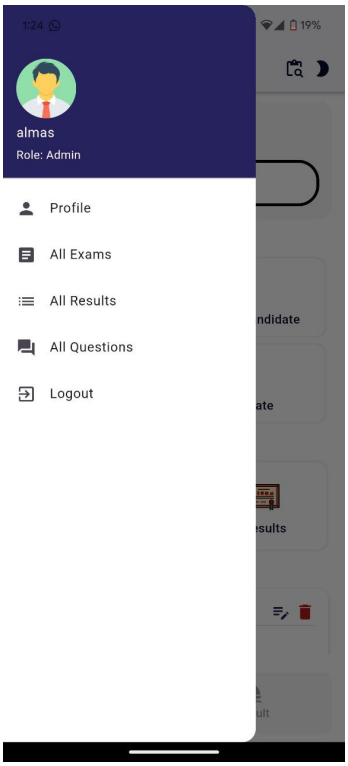


Figure 3.6: Side bar

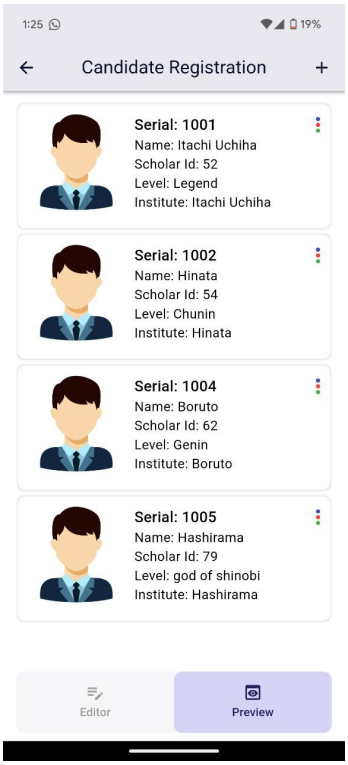


Figure 3.7: Candidate registration

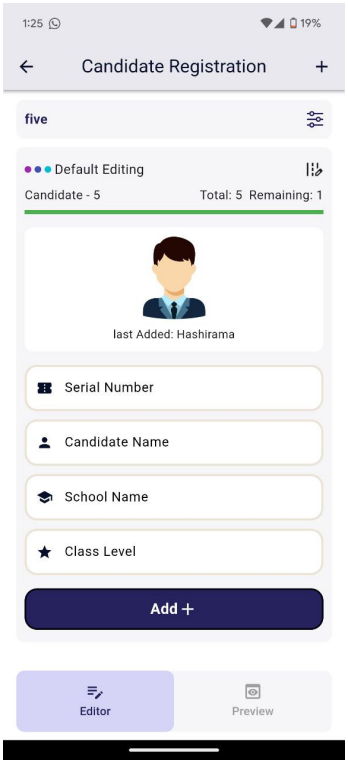


Figure 3.8: Candidate add page

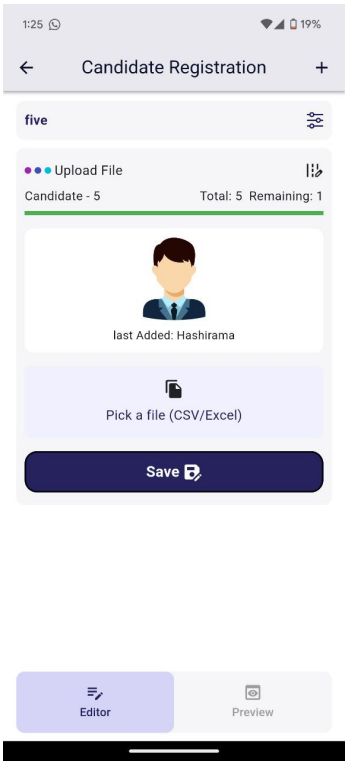


Figure 3.9: Candidate registration save page

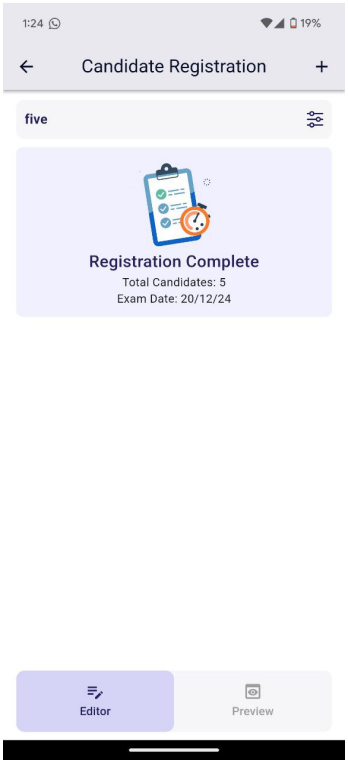


Figure 3.10: Registration confirm page

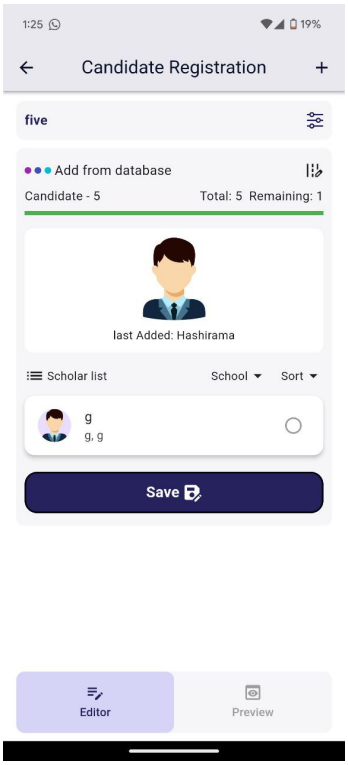


Figure 3.11: Registered candidate list

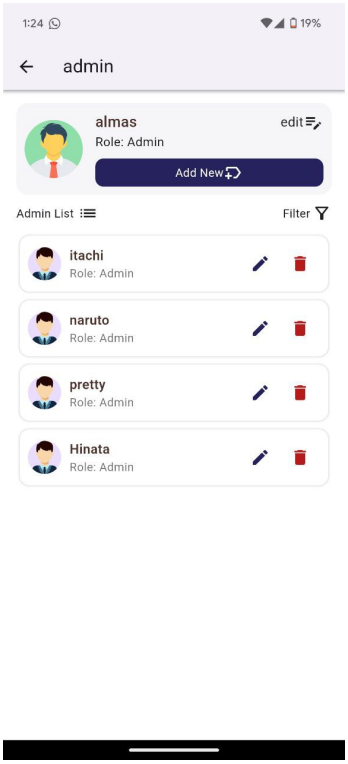


Figure 3.12: Admin list

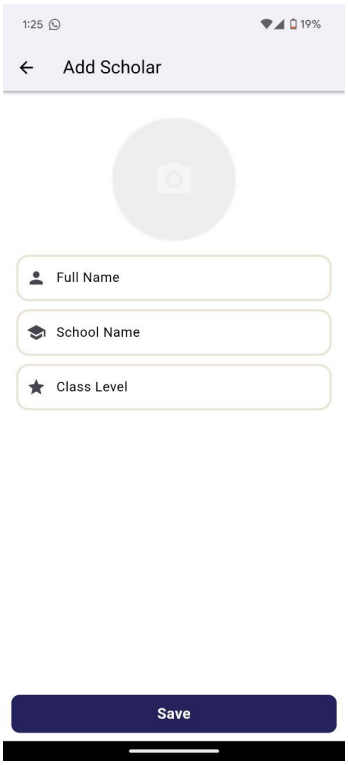


Figure 3.13: Add scholar page

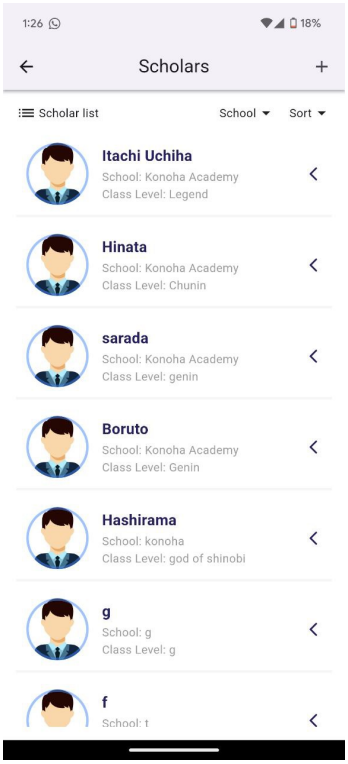


Figure 3.14: Scholar list

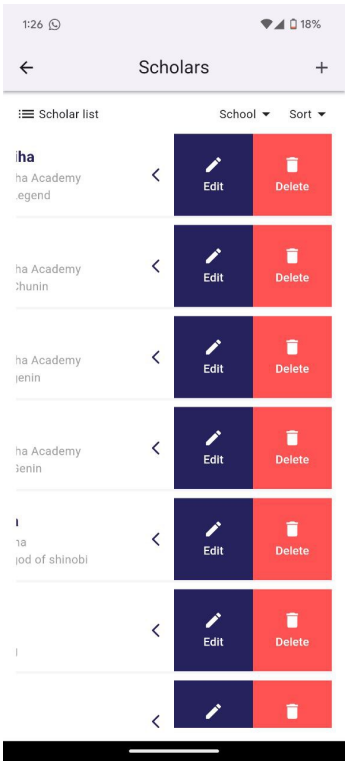


Figure 3.15: Scholar update

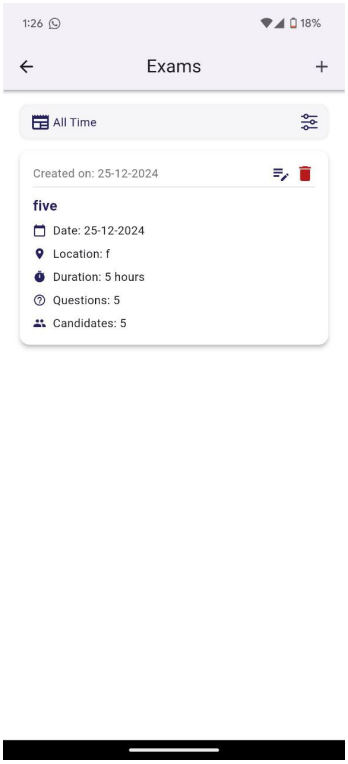


Figure 3.16: Exam details

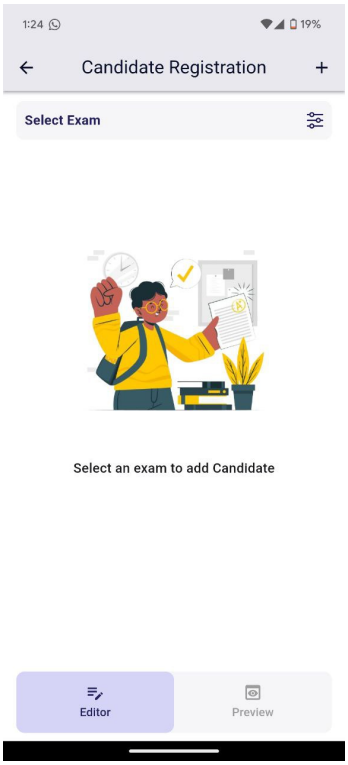


Figure 3.17: Create exam

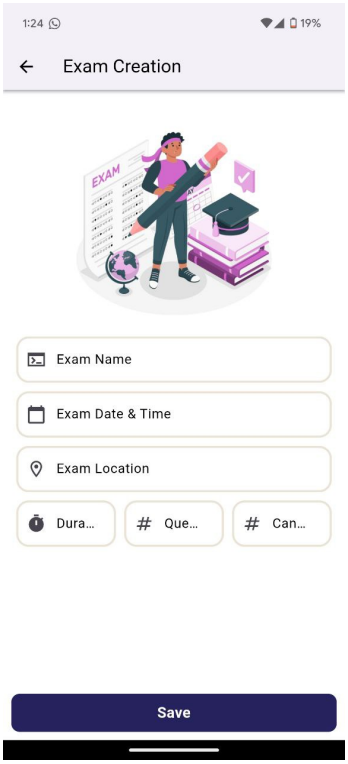


Figure 3.18: exam creation

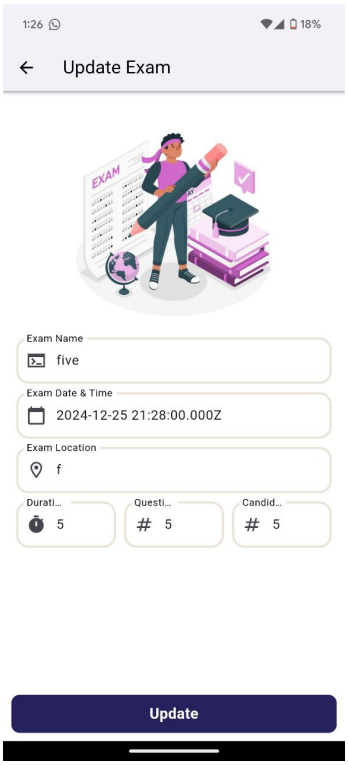


Figure 3.19: Update exam

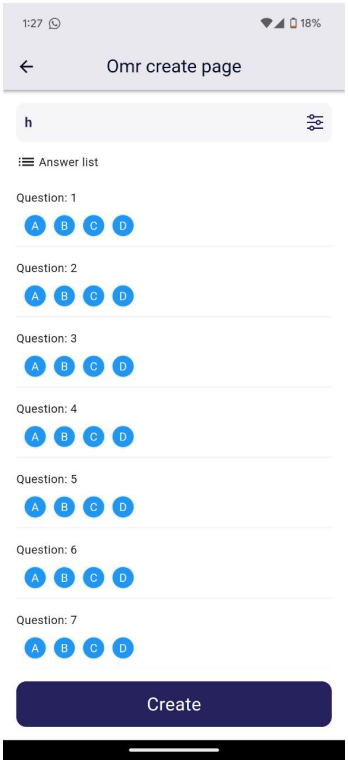


Figure 3.20: OMR create page

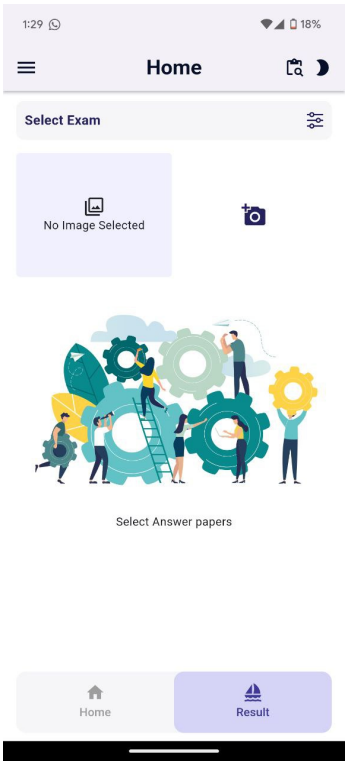


Figure 3.21: Select exam

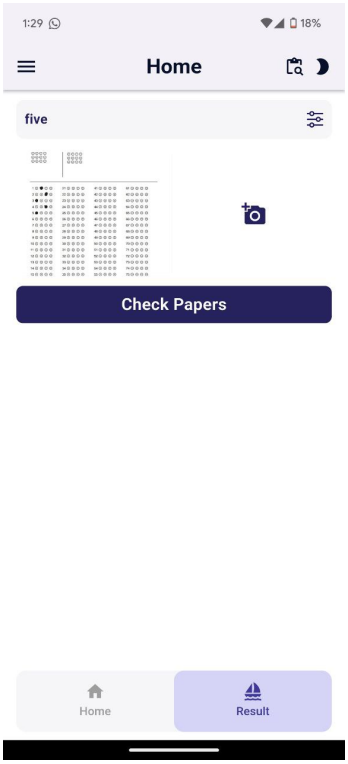


Figure 3.22: Check papers

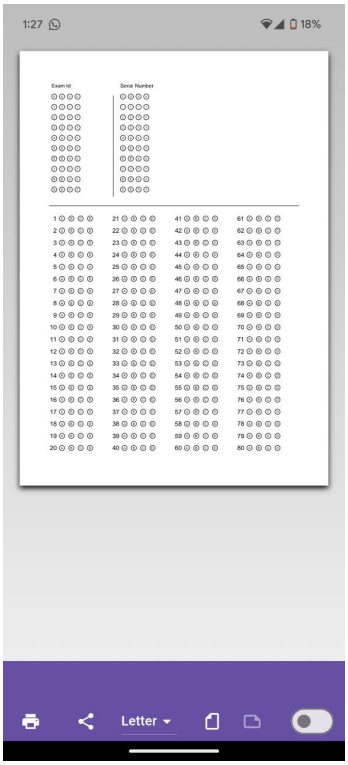


Figure 3.23: OMR sheet

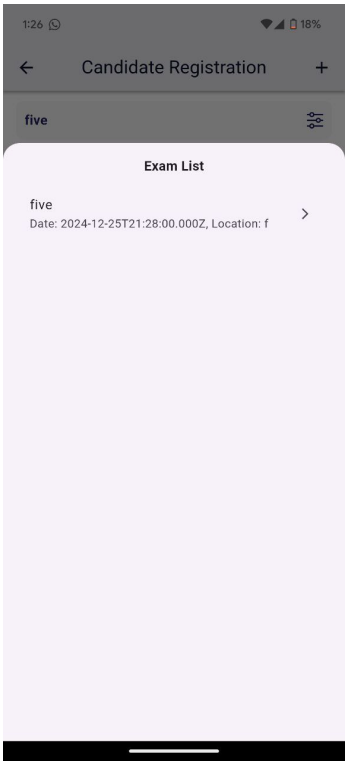


Figure 3.24: Exam filter

Figure 3.25: result list

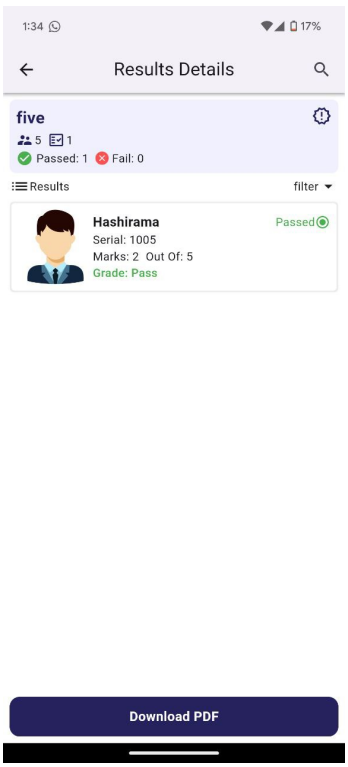


Figure 3.26: Result details

3.2 Detailed Methodology and Design

In developing the MCQ Exam Management System, several alternate solutions and approaches were considered to ensure optimal system performance, scalability, and usability. Below are the key considerations and the rationale behind the selected technologies and design choices.

Backend Development:

- **Express.js**: Express.js is known for its non-blocking, asynchronous nature, making it efficient in handling concurrent requests. It also allows rapid development of RESTful APIs, which is crucial for the system's interaction with the frontend and OMR library.
- **Python & OpenCV:** OpenCV (Open Source Computer Vision Library) is a powerful library for computer vision and image processing tasks. Combined with Python, it enables developers to perform a wide range of operations like image analysis, object detection, video processing, and more.

Frontend Development:

- **Flutter**: Flutter's widget-based structure and hot-reload feature allow for faster UI development, and its ability to run on both Android and iOS without additional code makes it highly scalable. The framework also provides robust support for integrating with REST APIs, making it ideal for this system.

OMR Integration:

- **Custom OMR Processing**: A custom solution was developed using available open-source libraries and tools for OMR processing. This approach provides greater flexibility, control over the accuracy of answer extraction, and integration with the backend for automatic grading.

Database Management:

- **Relational Database (MySQL)**: MySQL is reliable, well-documented, and supports ACID transactions, which ensures data integrity and security. This is critical for handling sensitive student data and exam results.

Security:

- **JWT**: A secure login system with JWT (JSON Web Tokens) for user authentication was selected. This approach provides stateless authentication, which is scalable and ensures secure access for both students and administrators.

3.3 Project Plan

The project will be developed in phases, with each phase focusing on specific deliverables. The following is the detailed timeline for the project:

Phase 1: Requirements Gathering and Analysis (1 week)

- Gather functional and non-functional requirements from stakeholders.
- Identify challenges in existing systems and define project scope.

Phase 2: System Design and Architecture (2 weeks)

- Design the overall system architecture, including backend and frontend components.
- Create database schemas and define API structure.
- Develop UI wireframes and user flow diagrams.

Phase 3: Development and Implementation (5 weeks)

- **Week 1-2**: Set up the backend with Express.js and integrate MySQL for database management.
- **Week 3-4**: Develop the frontend with Flutter and integrate with the backend APIs.
- **Week 5**: Implement OMR integration and automated grading features.

Phase 4: Testing and Debugging (2 weeks)

- Conduct unit testing for individual components.
- Perform integration testing to ensure proper interaction between modules.
- Conduct user acceptance testing with administrators and students to verify usability.

Phase 5: Deployment (1 week)

- Host the system on a secure cloud server.
- Set up server environments, including SSL encryption and database backups.
- Deploy the application and monitor for performance issues.

Phase 6: Maintenance and Updates (ongoing)

- Regularly update the system to fix bugs, enhance performance, and add new features.
- Monitor system usage, handle user feedback, and incorporate new requests for future versions.

3.4 Task Allocation

Given that this project is being developed as a solo endeavor, the following task allocation outlines the responsibilities I will manage throughout the project:

i. Backend Development (Express.js):

- Design the server architecture and set up Express.js to handle requests and responses.
- Develop and integrate REST APIs for managing exams, student data, and results.
- Set up authentication with JWT for secure user login.

ii. Frontend Development (Flutter):

- Design the UI for both administrators and students.
- Develop the functionality for students to take exams, view results, and interact with the system.
- Create the admin dashboard for managing exams, student data, and results.

iii. OMR Integration:

- Integrate an OMR library or tool for scanning and extracting answers from scanned sheets.
- Implement the logic for comparing answers with the answer key and calculating results.

iv. Database Management:

- Design and set up the MySQL database to store student records, exams, question banks, and results.
- Write SQL queries for data management and ensure data integrity through proper normalization.

v. Testing and Debugging:

- Conduct unit testing for individual modules (backend, frontend, database).
- Perform integration testing to ensure seamless interaction between the system's components.
- Address any issues or bugs found during testing.

vi. Deployment and Maintenance:

- Set up the production environment and deploy the application to a secure cloud server.
- Monitor system performance and apply patches or updates as necessary.

3.5 Summary

This chapter includes features of system design for the MCQ Exam Management System. It involves several key components, including a backend for handling exam creation, student management, and automated grading using Optical Mark Recognition (OMR) technology, alongside a user-friendly frontend interface. The design incorporates a centralized relational database to securely store student data, exam details, and results. The designed system ensures and secures users' data with data encryption and user authentication. Overall, the system aims to simplify the exam management process, enhance grading accuracy, and provide a seamless experience for administrators.

Chapter 4

Implementation and Results

This chapter describes the implementation of the MCQ Exam Management System, including the setup of the development environment, the evaluation of system functionality, and a discussion of the results. The system was designed to automate MCQ exam management processes with features like Optical Mark Recognition (OMR) and centralized result storage.

4.1 Environment Setup

This section outlines the technical environment used to develop the system:

- **Frontend:** Developed using **Flutter** with **Dart**, providing a cross-platform user interface for administrators and candidates.
- **Backend:** Built with **Express.js** using **JavaScript**, ensuring efficient handling of API requests and database operations.
- **Database:** A **Mysql** database is used to store question banks, user data, test details, and exam results securely.
- **OMR Integration:** Implemented with opencv library for image processing and marked answers extraction, ensuring accurate answer recognition.
- **Development Tools:** The project was developed using Visual Studio Code, with version control managed via GitHub.

4.1.1 API Design

Domain: api.rahatalmas.com
RESTful APIs

USER
GET: /api/user/list

GET: /api/user/list/:id
POST: /api/user/register
POST: /api/user/login
PUT: /api/user/update/name
PUT: /api/user/update/password
PUT: /api/user/update/role
DELETE: /api/user/delete/:id

EXAM
GET: /api/exam/list
GET: /api/exam/list/:examId
POST: /api/exam/add
PUT: /api/exam/update
DELETE: /api/exam/delete/:id

SCHOLAR
GET: /api/scholar/list
GET: /api/scholar/filteredlist/:examlist
POST: /api/scholar/add
PUT: /api/scholar/update
DELETE: /api/scholar/delete/:id

CANDIDATE
GET: /api/candidate/:examId
GET: /api/candidate/count/:examId
POST: /api/candidate/add
DELETE: /api/candidate/delete/:serialNumber/:examId

RESULT
GET: /api/result/allresult
GET: /api/result/all/:examId
GET: /api/result/all/myresult/:examId/:serialNumber
POST: /api/result/add
PUT: /api/result/update
DELETE: /api/delete/:examId
DELETE: /api/delete/individual/:examId/:serialNumber

ANSWER
GET: /api/answer/all/:examId POST: /api/answer/add
PUT: /api/answer/update

- **IDE Setup & File Structure**

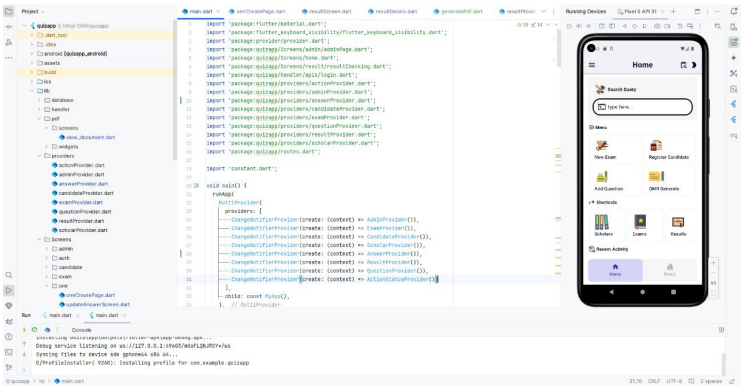


Figure 4.1: Android Studio setup and Flutter File structure

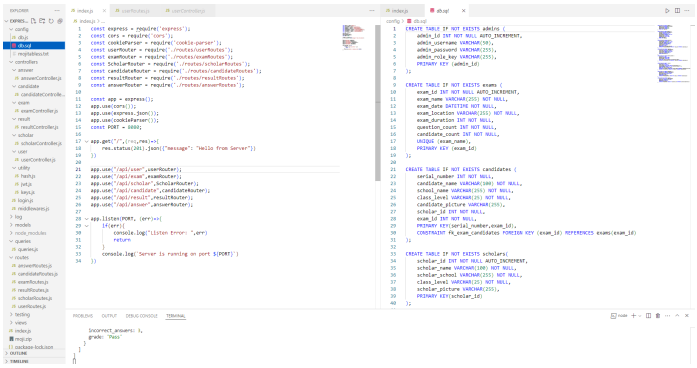


Figure 4.2: ExpressJS server code and file structure

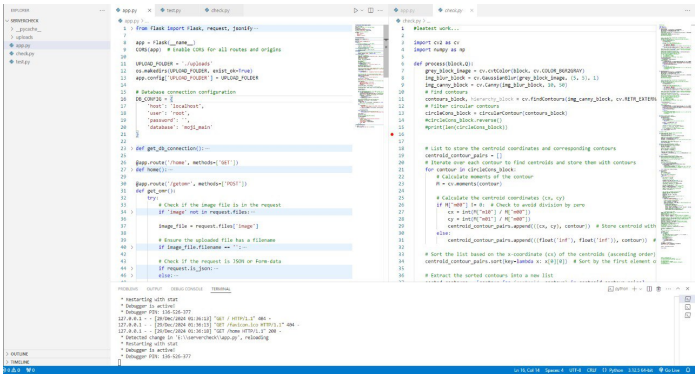


Figure 4.3: Flask server code and file structure

- **Database**

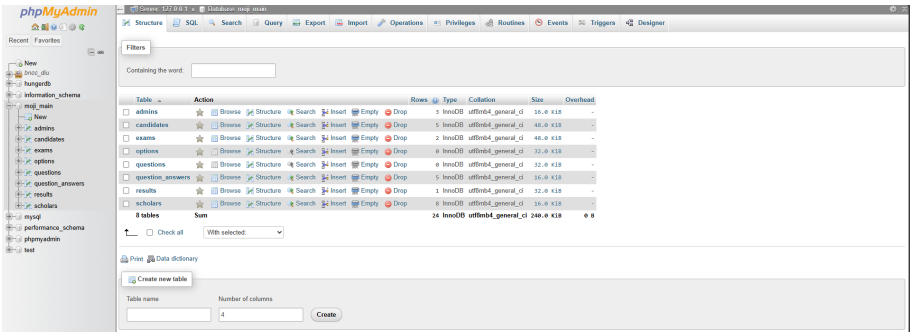


Figure 4.4: Database

- **API Testing & Response**

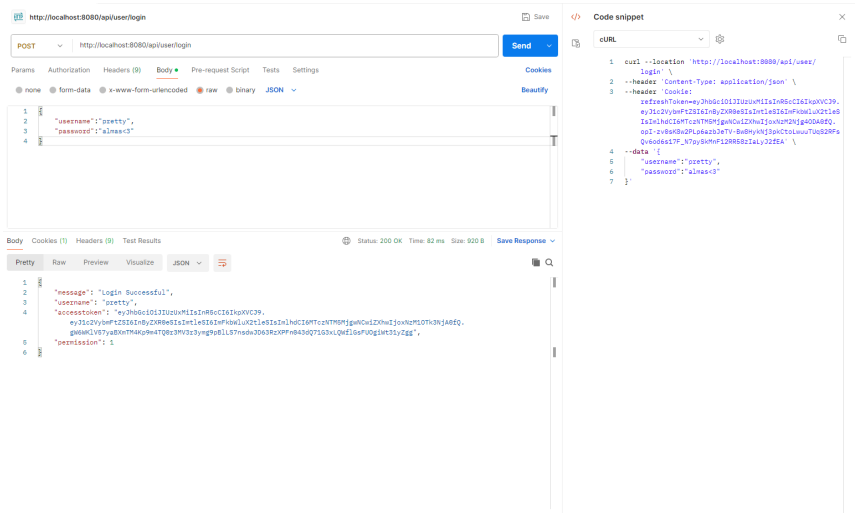


Figure 4.5: Login API

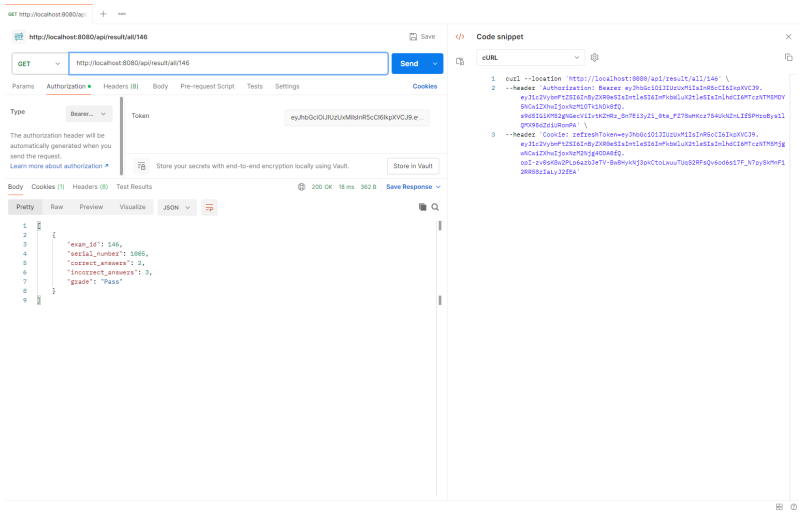


Figure 4.6: specific person result

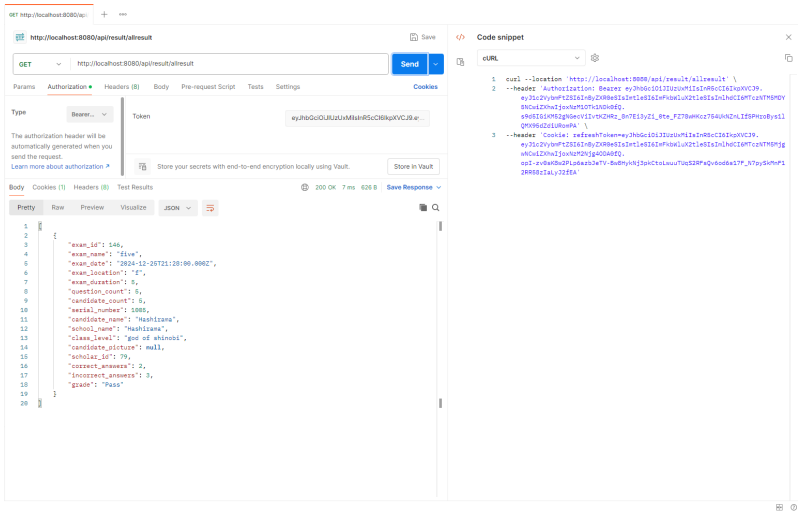


Figure 4.7: all result for a exam

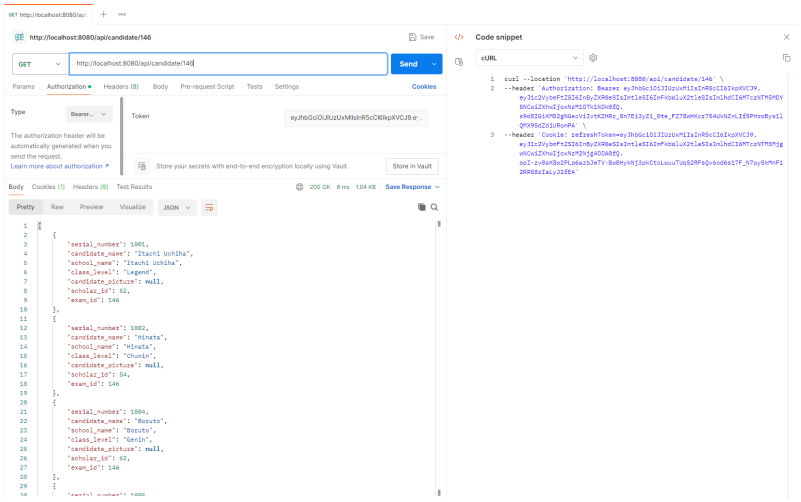


Figure 4.8: Exam candidates

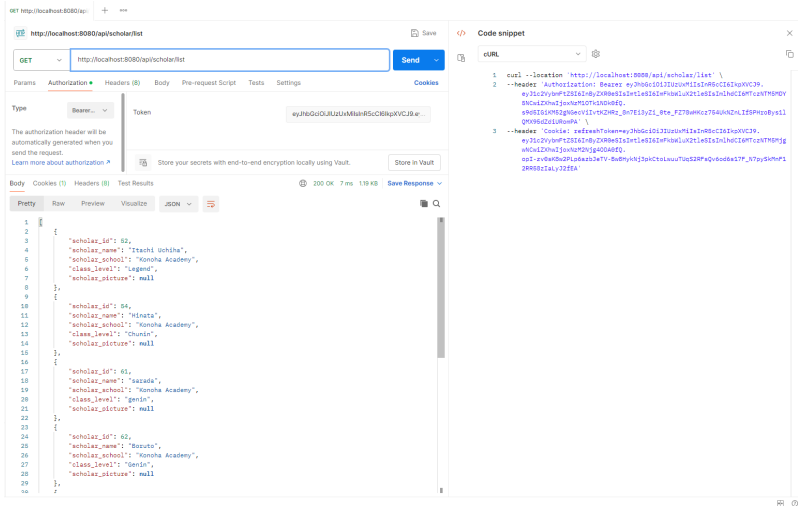


Figure 4.9: Scholar List

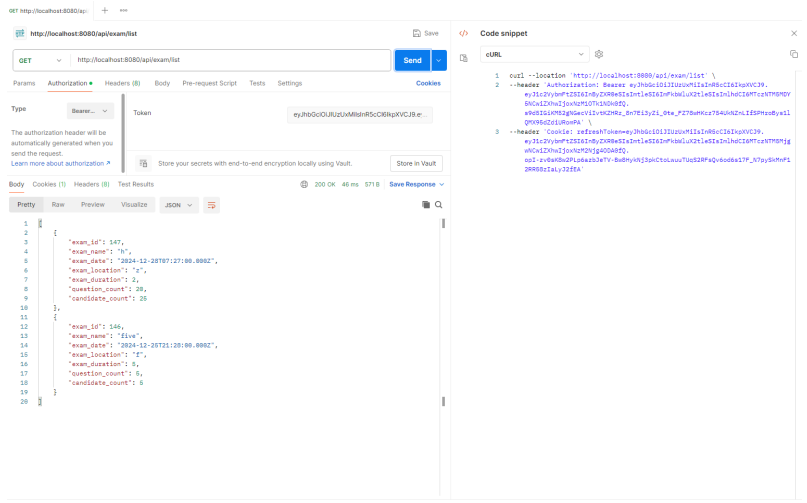


Figure 4.10: Exam List

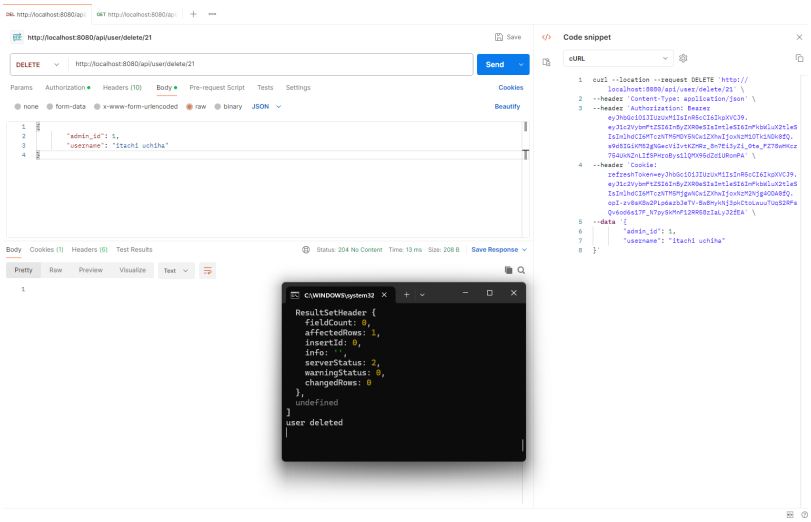


Figure 4.11: User Delete

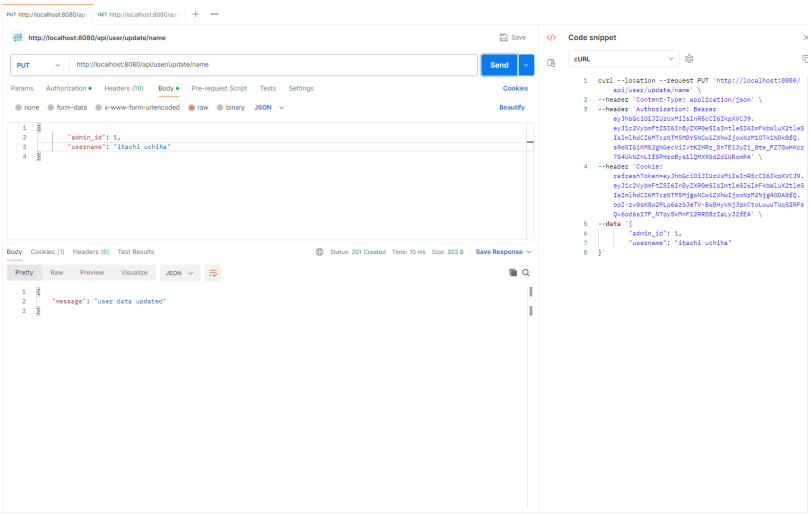


Figure 4.12: User Data Update

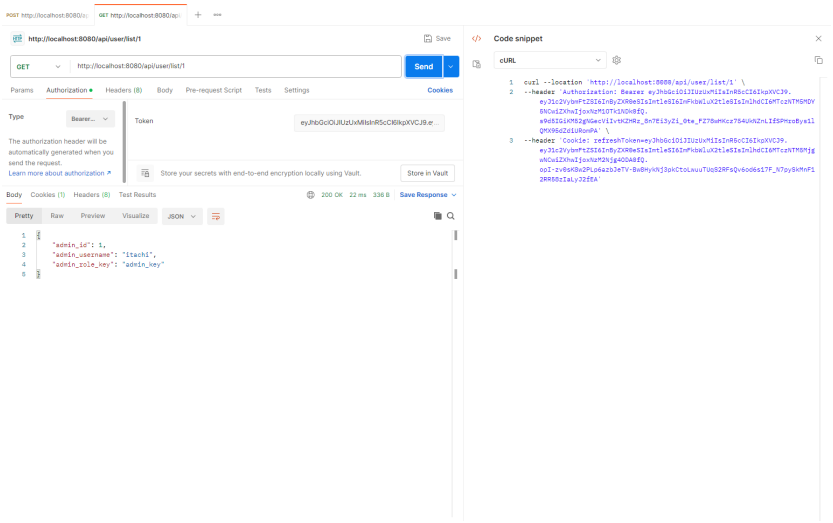


Figure 4.13: User data

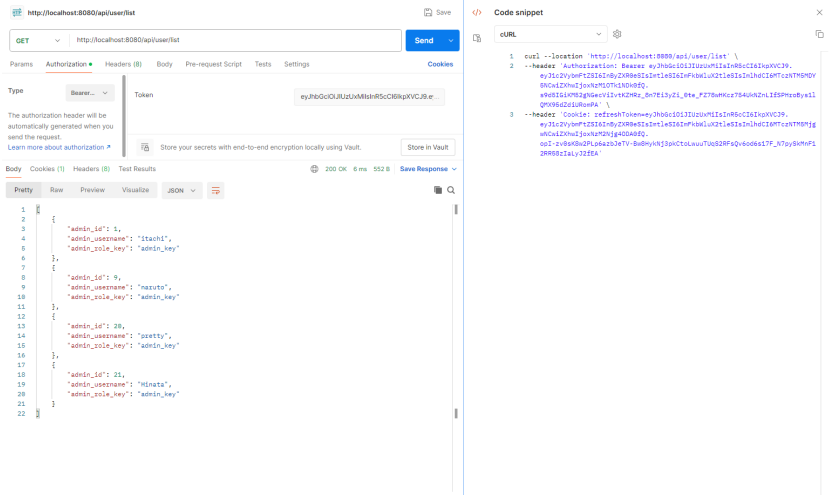


Figure 4.14: User List

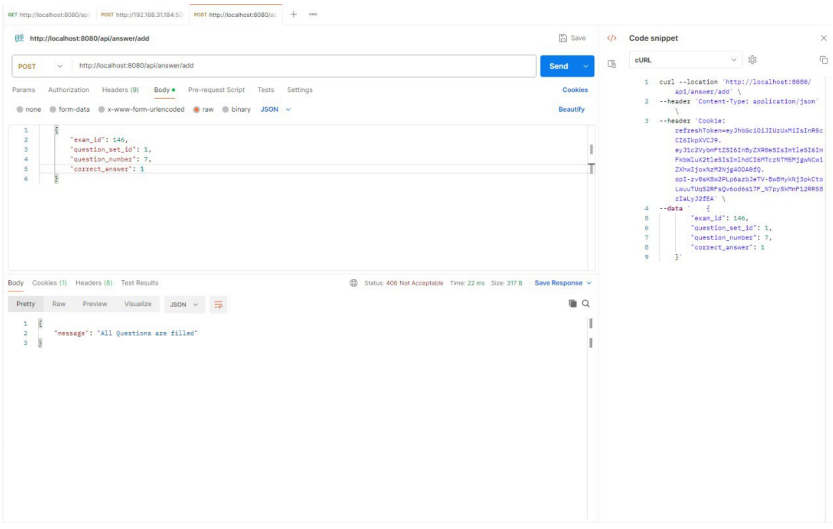


Figure 4.15: Add answer

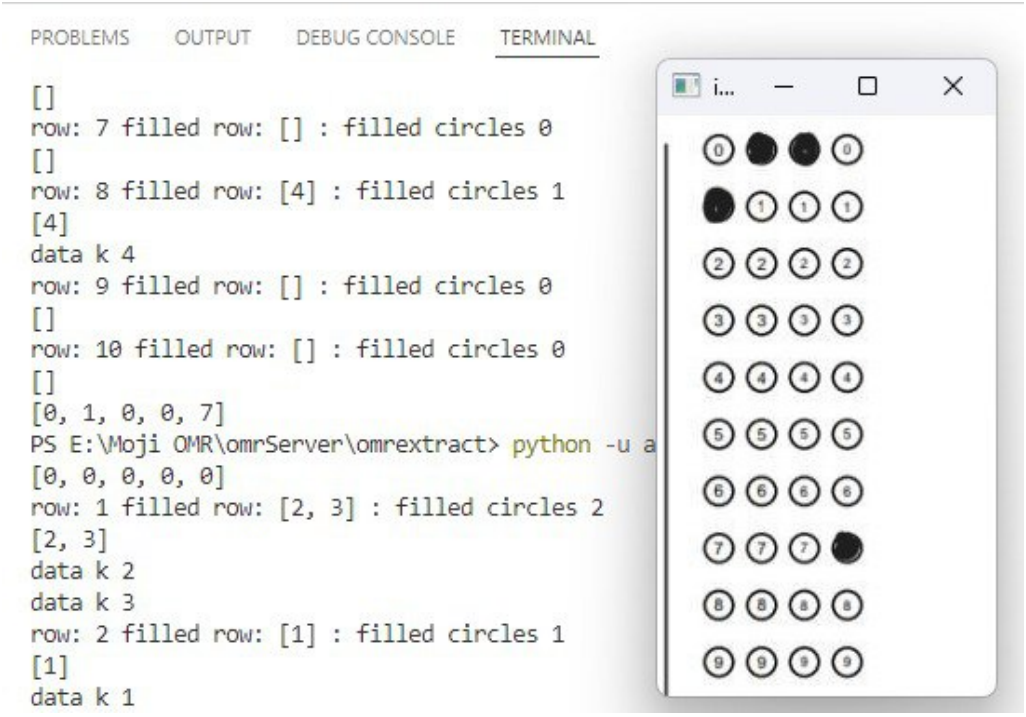


Figure 4.16: OMR to Serial Number Extraction

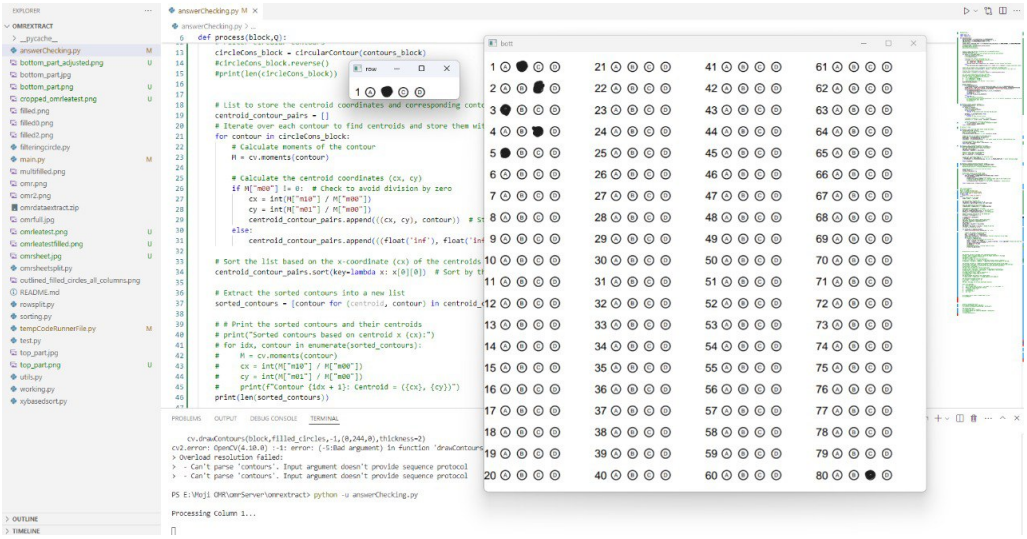


Figure 4.17: OMR to result Extraction

4.2 Results and Discussion

Present the outcomes and insights:

- **System Features Performance:** All core functionalities were successfully implemented. The OMR feature significantly reduced the manual workload by automating grading and result generation.
- **Challenges:** Low-quality OMR sheets led to minor recognition errors, which were mitigated by preprocessing images.
- **Insights:** The integration of OMR technology streamlined the grading process, and the PDF export feature enhanced result sharing.

4.3 Summary

This chapter presented the implementation and results of the MCQ Exam Management System. The integration of Express.js, Flutter, and OMR technology achieved the objectives of automating MCQ exam processes. The system demonstrated high efficiency, accuracy, and usability, making it a valuable solution for managing large-scale exams.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

The MCQ Exam Management System adheres to various industry standards to ensure reliability, compatibility, and security.

5.1.1 Software Standards

The development of this application adheres the software standards to ensure quality, compatibility, and security.

- Coding Standards: Followed best practices in coding, such as clean code principles, proper documentation, and naming conventions for variables, classes, and functions.
- Testing Standards: Utilized unit testing, stress testing and integration testing methods to ensure the functionality and robustness of the system.
- Security Standards: Implemented Idor, broken access control, CSRF, SSRF, secure authentication and data encryption to comply with Hash algorithm and other data protection regulations.
- User Interface Standards: Designed a user friendly interface with consistent typography, navigation, and layout for accessibility compliance.
- Performance Standards: Ensured efficient database queries and response times to support large scale exam management.

5.1.2 Hardware Standards

The system requires the following hardware to operate efficiently. Server Requirements: Processor: Minimum Quad-Core 2.5 GHz RAM: Minimum 16 GB Storage: SSD with at

5.1.3 Communication Standards

The communication between the client and server and between system components adheres to the following protocols: Network Protocols: HTTPS for secure data transmission. Data Communication: JSON formats for structured data exchange between the web interface and the backend. Error Logging: Integrated logging frameworks such as Express JS for efficient error reporting and troubleshooting. API Standards: RESTful API design to ensure modularity and interoperability for future enhancements.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The MCQ Exam Management System improves the overall experience for educators, and administrators. Educators and administrators benefit from a streamlined workflow, allowing them to focus more on teaching and organizational development rather than manual grading and result management. Institutions adopting this system also see enhanced efficiency, reduced operational overhead, and better stakeholder satisfaction, creating a positive impact on all participants in the educational ecosystem.

5.2.2 Impact on Society & Environment

The system drives societal progress by integrating advanced technology into education, enabling a more efficient learning environment. By encouraging transparency and efficiency, it fosters trust among educators, and stakeholders. Environmentally, the system reduces the reliance on paper by digitizing exams and results, leading to a substantial decrease in paper waste. Additionally, it eliminates the need for transportation and storage of physical exam materials, lowering the associated carbon footprint and contributing to a more sustainable future.

5.2.3 Ethical Aspects

The system upholds strong ethical standards by prioritizing data security and privacy. It ensures compliance with global data protection regulations, safeguarding the personal information of students and educators. By automating the evaluation process, it eliminates biases and inconsistencies that may arise from manual grading, promoting fairness and equity.

5.2.4 Sustainability Plan

- **Long Term Scalability:** I have currently hosted on local server or device to handle an increasing number of users and exams without hardware overhauls. Regular system updates to remain compatible with new technologies and standards.

- **Energy Efficiency:** Optimizing server performance to reduce energy consumption and encouraging the use of energy efficient devices for OMR scanning and data processing.
- **User Training:** Conducting workshops for administrators and teachers to ensure effective system use, reducing operational inefficiencies.
- **Maintenance and Support:** I have established a dedicated support team for troubleshooting and updates.

5.3 Project Management and Financial Analysis

The development of the MCQ Exam Management System was divided into several phases to ensure systematic progress.

- Requirement Analysis: Identification of key features and stakeholder expectations.
- System Design: Development of the architecture, database schema, and user interface mockups.
- Development: Implementation of features such as OMR processing, exam management, and result generation.
- Testing: Rigorous testing of functionalities, including unit testing, stress testing.
- Deployment: Hosting the system on a scalable and secure server infrastructure.
- Feedback: Gathering feedback from users.

Table 5.1: Project timeline & financial analysis

Tasks	Time	Estimated Cost
Project Planning	1 Week	90 USD
Requirement Analysis and UI Design	2 Week	100 USD
App development	3-5 Weeks	600 USD
Database Design	6-7 Weeks	200 USD
API development	8-9 Weeks	500 USD
Testing and Debugging	10 Week	200 USD

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

In this section, a mapping of the problem and solutions are provided with targeted Program Outcomes (PO's).

Table 5.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	EP2 Range of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent of Stakeholder Involvement	EP7 Interdependence
√	*X*	√	√	*X*	√	*X*

Mapping with Knowledge Profile for EP1

This table 5.3 is designed to map the EP1 to the Knowledge Profile.

Table 5.3: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
√	√	√	√	√

5.4.2 Engineering Activities

In this section, a mapping with engineering activities has been provided.

Table 5.4: Mapping with complex engineering activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
√	√	√	√	√

5.5 Summary

This chapter covers several key aspects of the MCQ Exam Management System, focusing on the standards, societal impact, and financial management. It includes the software and hardware standards essential for ensuring efficient system performance and the communication standards required for smooth data transmission. The system's impact on life,

society, and the environment is highlighted, emphasizing its role in improving educational efficiency, reducing administrative workload, and minimizing the environmental footprint by cutting down paper usage. Ethical considerations are also addressed, ensuring data privacy, fairness in grading, and equal access. A sustainability plan is outlined to maintain and scale the system, utilizing cloud-based infrastructure and open-source tools. The chapter concludes with a detailed financial analysis, presenting both initial and alternate budget plans, with the alternate budget offering a more cost-effective approach.

Chapter 6

Conclusion

6.1 Summary

The MCQ Exam Management System offers an innovative solution to automate and streamline the process of administering multiple-choice exams, from creating and managing exams to grading and generating results. With features such as Optical Mark Recognition (OMR) technology for automated answer sheet evaluation, centralized storage of exam data, the system significantly reduces the administrative workload and improves efficiency. The project has been designed with a focus on scalability, usability, and cost-effectiveness, ensuring that it can meet the demands of educational institutions of various sizes.

6.2 Limitation

Despite the system's many advantages, there are some limitations. The accuracy of the OMR technology depends heavily on the quality of the scanned answer sheets, which could pose challenges if the sheets are not filled out correctly or are poorly scanned. Additionally, while the system provides automated grading, it may not be suitable for complex question types, such as essay questions, which require subjective evaluation. The system also depends on stable internet and cloud-based infrastructure, which may not be accessible in all regions, potentially limiting its use in certain areas with less reliable internet access.

6.3 Future Work

Future work for the MCQ Exam Management System includes integrating more advanced features, such as machine learning algorithms for adaptive testing, which would allow the system to tailor exams to individual students' knowledge levels. Expanding the system to handle other types of assessments, such as essays and project submissions, would make it a more comprehensive solution for diverse educational needs. Additionally, further

optimization for mobile platforms and improving the OMR technology to handle a wider variety of answer sheet formats could enhance its usability and accessibility. Finally, expanding the system's reach to global markets and offering multilingual support could make the system a valuable tool for educational institutions worldwide.

References

[1] *Evalbee (OMR Sheet Scanner) - apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=com.ekodroid.omrevaluator (Accessed: 12 December 2024).

[2] *OMR D̃ynamic OMR sheet - apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=com.reakabc.omr (Accessed: 12 December 2024).

[3] *Live MCQTM - apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=com.livemcq.livemcq (Accessed: 12 December 2024).

[4] *Aspose.OMR – create PDF sheets - apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=cloud.aspose.omr.app (Accessed: 12 December 2024).

[5] *Brritto: Learn and Test - apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=tech.innospace.brritto (Accessed: 12 December 2024).

[6] *OMR - Apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=com.famesun.omr (Accessed: 12 December 2024).

[7] *MCQS - apps on Google Play* (no date) *Google*. Available at: https://play.google.com/store/apps/details?id=com.mcqs.anita.mcqs_android_coding (Accessed: 12 December 2024).

Mobile app for MCQ Exam Management and Results Generation

ORIGINALITY REPORT

7%	5%	0%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to United International University Student Paper	3%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	Submitted to University of Wales Institute, Cardiff Student Paper	1%
4	ijrpr.com Internet Source	<1%
5	Submitted to Leeds Beckett University Student Paper	<1%
6	www.coursehero.com Internet Source	<1%
7	www.imperial.ac.uk Internet Source	<1%
8	v1.overleaf.com Internet Source	<1%