Table of Contents

[1. Introduction 2](#_Toc15303113)

[1.1. Existing System 2](#_Toc15303114)

[1.2. Need for the New System 2](#_Toc15303115)

[1.3. Objective of the New System 2](#_Toc15303116)

[1.4. Problem Definition 2](#_Toc15303117)

[1.5. Core Components 2](#_Toc15303118)

[1.6. Project Profile 2](#_Toc15303119)

[1.7. Assumptions and Constraints 2](#_Toc15303120)

[1.8. Advantages xand Limitations of the Proposed System 2](#_Toc15303121)

[1.9. Proposed Time Line Chart 3](#_Toc15303122)

[2. Requirement Determination & Analysis 3](#_Toc15303123)

[2.1. Requirement Determination 3](#_Toc15303124)

[2.2. Targeted Users 3](#_Toc15303125)

[3. System Design 3](#_Toc15303126)

[3.1. Use Case Diagram 3](#_Toc15303127)

[3.2. Class Diagram 4](#_Toc15303128)

[3.3. Interaction Diagram 5](#_Toc15303129)

[3.4. Activity Diagram 7](#_Toc15303130)

[3.5. Data Dictionary 8](#_Toc15303131)

[4. Development 10](#_Toc15303132)

[4.1. Coding Standards 10](#_Toc15303133)

[5. Agile Documentation 10](#_Toc15303134)

[5.1. Agile Project Charter 10](#_Toc15303135)

[5.2. Agile Project Charter 10](#_Toc15303136)

[5.3. Agile Project Plan 10](#_Toc15303137)

[5.4. Agile User Story ( Minimum 3 Tasks) 10](#_Toc15303138)

[5.5. Agile Release Plan 10](#_Toc15303139)

[5.6. Agile Sprint Backlog 10](#_Toc15303140)

[5.7. Agile Test Plan 10](#_Toc15303141)

[5.8. Earned-value and burn charts 10](#_Toc15303142)

[6. Proposed Enhancements 10](#_Toc15303143)

[7. Conclusion 10](#_Toc15303144)

[8. Bibliography 11](#_Toc15303145)

# Introduction

***The Exam Management System for Universities is like a powerful online tool that helps schools and colleges handle exams smoothly. It's built using a combination of technologies (MERN stack) to make it reliable and efficient. This system is carefully designed to make it easy for universities to create, schedule, conduct, and manage exams without any hassle.***

***In today's education world, where exams are really important to understand how much students know, having a smart and versatile Exam Management System is crucial. This system simplifies the whole process of organizing and handling exams, making it a lot easier for everyone involved.***

## Existing System

***Think of the way universities used to handle exams - lots of manual paperwork, creating problems and slowing down the process. The old method, with all its history, has a bunch of issues. So, it's time to bring in something more modern and tech-savvy.***

***In the current systems, dealing with physical exam papers creates a bunch of complications. From printing and handing them out to collecting them, it takes up a lot of resources and often leads to mistakes and delays. Coordinating schedules manually just adds more problems, like clashes and issues with assigning rooms. It's time to ditch those old ways and make everything much simpler!***

## Need for the New System

***In light of the challenges posed by traditional manual and paper-based exam management systems, there is an imperative need for a modernized approach that aligns with the demands of contemporary educational landscapes. The limitations inherent in the existing systems underscore the necessity for a comprehensive and technologically advanced solution, prompting the development of the Exam Management System for Universities.***

## Objective of the New System

***Imagine the Exam Management System for Universities as a clever upgrade to how exams are usually handled. It's like a super-smart tool that aims to make tasks like creating and scheduling exams quicker and easier, saving time and resources. Instead of dealing with physical papers, everything goes digital, ensuring information is secure and risks are minimized.***

***The focus is on being open and easy for everyone to use. Administrators, professors, and students can all access information in real-time, making collaboration a breeze. Results are processed fast, providing quick feedback for a more dynamic learning experience. The system also uses smart analytics to help universities make better decisions based on data.***

***It's designed to be user-friendly, with a simple interface that's easy for everyone to navigate. Plus, it's flexible and can adapt to the university's changing needs and new technology. Basically, it's here to make managing exams better for students and the whole institution.***

## Problem Definition

***The existing manual and paper-based exam management systems in universities result in time-consuming processes, resource inefficiencies, and security vulnerabilities. Manual scheduling often leads to clashes, and result processing is labor-intensive, causing delays in feedback. These challenges highlight the need for the Exam Management System, aiming to streamline processes, enhance security, and provide real-time insights for more efficient examination management.***

## Core Components

**Student Registration System**

The Student Registration System enables secure user authentication, profile creation, and course/exam enrollment. This core component ensures accurate student records, seamless module integration, and overall system effectiveness.

**Student Login System**

The Login System is a crucial component facilitating secure access to the Exam Management System. It provides user authentication, ensuring a protected entry point for administrators, professors, and students. This core element is integral for maintaining system security, personalized user experiences, and efficient information retrieval within the broader framework of the Exam Management System.

**Registration System**

The Teacher Registration System helps educators manage their credentials digitally. Exam management ensures exams are fair and well-organized, including scheduling, paper creation, and grading, crucial for maintaining academic standards and evaluating student learning.

**Teacher Login System**

In the exam management system, the Teacher Login Model lets teachers log in securely. Once logged in, they can create exams, manage questions, and check students' results. It helps teachers handle exams smoothly and keep track of how students are doing.

**Exam Taking and Submission**

Exam Taking and Submission: This core component enables students to access and attempt assigned exams. It provides an interface for answering exam questions and submitting responses within the allocated time frame. Once completed, students submit their exam responses through the system for evaluation.

**Results and Feedback**

Results and Feedback: After the exam submission, this component generates results based on the evaluation of student responses. Additionally, it allows instructors to provide feedback on students' performance. Students can view their results and feedback to understand their strengths and areas needing improvement.

**Complaint Management**

Complaint Management: In case of any issues or discrepancies during the exam process, this component provides a platform for students to raise complaints. Students can report problems such as technical issues or unfair exam conditions, ensuring a fair and transparent assessment environment.

**Exam Creation and Assignment**

Exam Creation and Assignment: This core component empowers instructors to create new exams using a variety of question types. Once exams are created, instructors can assign them to individual students or groups based on course requirements and student capabilities.

**Student Management**

Student Management: This component facilitates the management of student profiles within the system. It includes functionalities such as adding new students, updating student information, and monitoring student progress and performance.

**Exam Management**

Exam Management: This component oversees the overall management of exams within the system. It includes functionalities such as organizing exam schedules, setting exam parameters (e.g., duration, passing score), and monitoring exam progress and completion status.

## Project Profile

**Project Profile**

Project Title **: Exam Management System**

Project Developer **: Ahmed Fahzain (En. No. : 21018501210001)**

**Saiyed Aasim (En. No. :21018501210055)**

**Sheth Mahamad Sarjil (En. No. : 21018501210061)**

Hardware Require**: Pentium 4 Micro Processor or Above**

**RAM 1 GB or Above**

**Hard Disk 60 GB or Above**

Software Require**: Platform : Window XP or Above**

**Front End : ReactJS**

**Back End : NodeJS**

**Framework : MERN Stack**

**Other Tools : MS Word 2013 (Documentation)**

**MS Powerpoint (Presentation)**

**Dia (Diagram)**

**Snipping Tool (ScreenShot)**

Project Guide : **Prof. Sangeeta Rajole**

## Assumptions and Constraints

***Assumptions:***

***1. Users have access to suitable devices for exam management.***

***2. Users are proficient in basic computer operations.***

***3. Security measures are in place to protect exam data integrity.***

***Constraints:***

***1.If one division exam is set on time no other exam should not be scheduled at that time of that division.***

***2. If a student skip a question so that their marks of that question should not be count.***

## Advantages and Limitations of the Proposed System

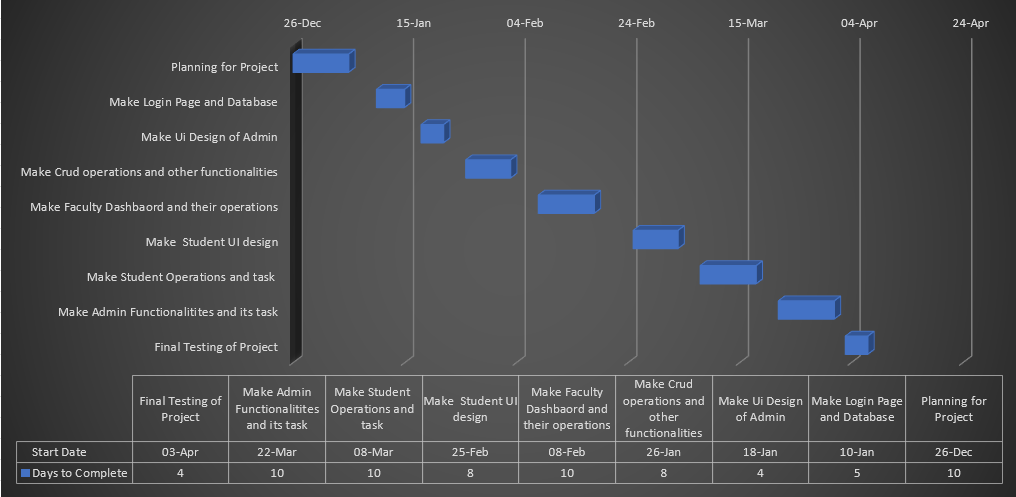
**Advantages of the Proposed System:**

1. **Efficiency Enhancement:**
   * The Exam Management System automates exam processes, reducing manual efforts and saving time for administrators, professors, and students. This efficiency enhancement contributes to a smoother and more streamlined examination workflow.
2. **Improved Security Measures:**
   * By transitioning from physical exam papers to a secure digital platform, the proposed system enhances the security and confidentiality of exam content. This not only safeguards the integrity of examinations but also minimizes risks associated with traditional paper-based systems.
3. **Real-time Information Access:**
   * Providing real-time access to exam-related information for administrators, professors, and students fosters transparency and collaboration.

**Limitations of the Proposed System:**

1. **Dependency on Technology Infrastructure:**
   * The success of the Exam Management System relies on the availability and proper functioning of technology infrastructure. Any disruptions or inadequacies in servers, databases, or network components may impact system performance.
2. **Initial Data Migration Challenges:**
   * The migration of existing data from manual systems to the new platform may pose challenges. Ensuring seamless data transfer without loss or corruption requires meticulous planning and execution, potentially causing temporary disruptions..
3. **User Training and Adoption:**
   * The system's effectiveness is contingent on user adoption and proficiency. Challenges may arise during the initial stages as administrators, professors, and students adapt to the new platform. Comprehensive training and support are crucial to mitigate potential hurdles.

## Proposed Time Line Chart

******

# Requirement Determination & Analysis

## Requirement Determination

1. **Collect information from the faculty members regarding the specific types of systems and functionalities they require.**

## Targeted Users

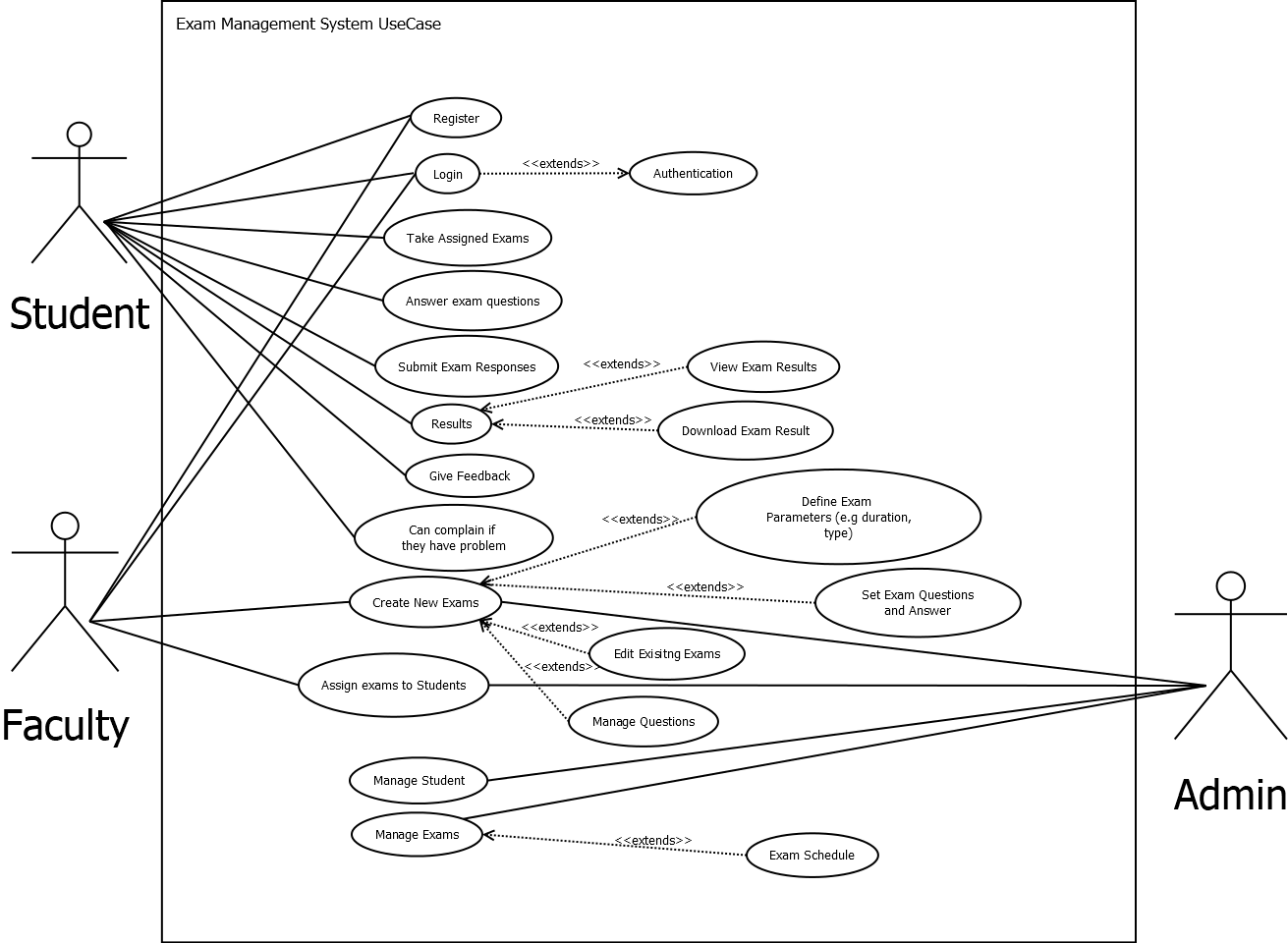
***1) ADMIN***

***2) FACULTY***

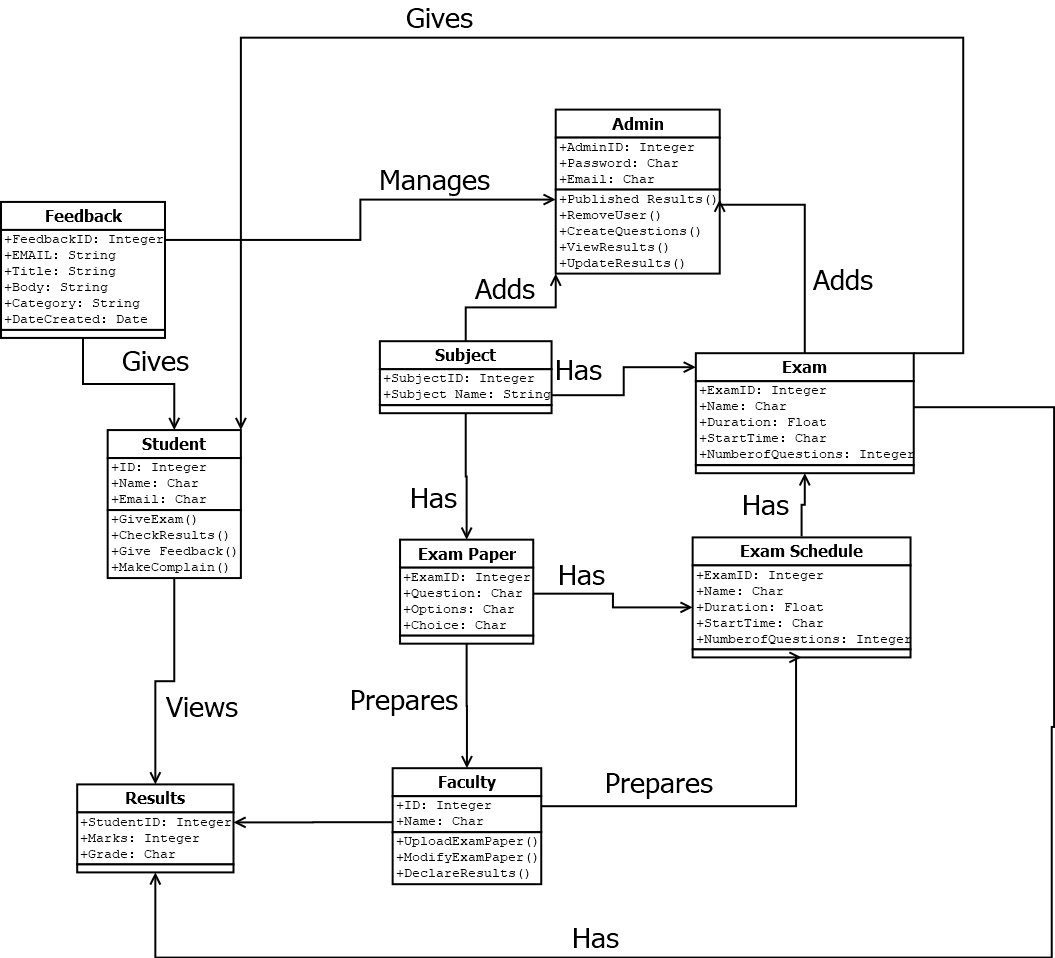
***3) STUDENTS***

# System Design

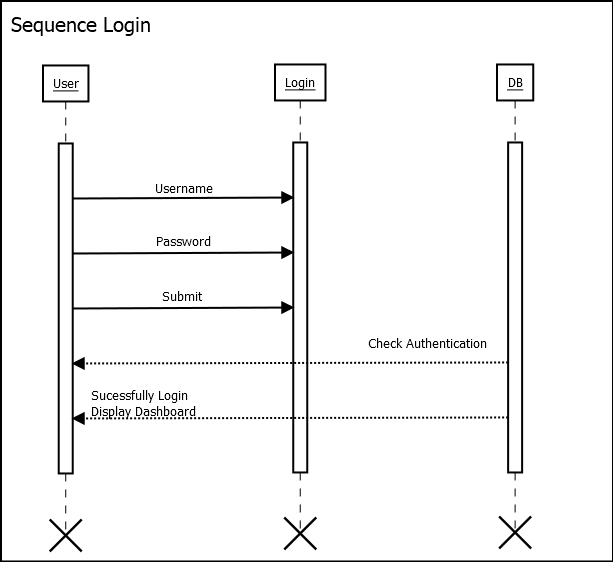
## Use Case Diagram

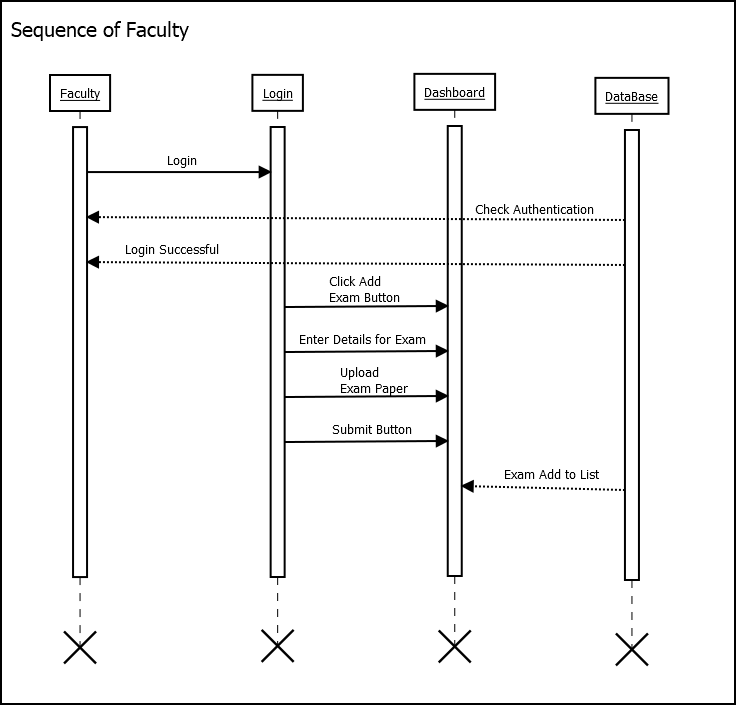


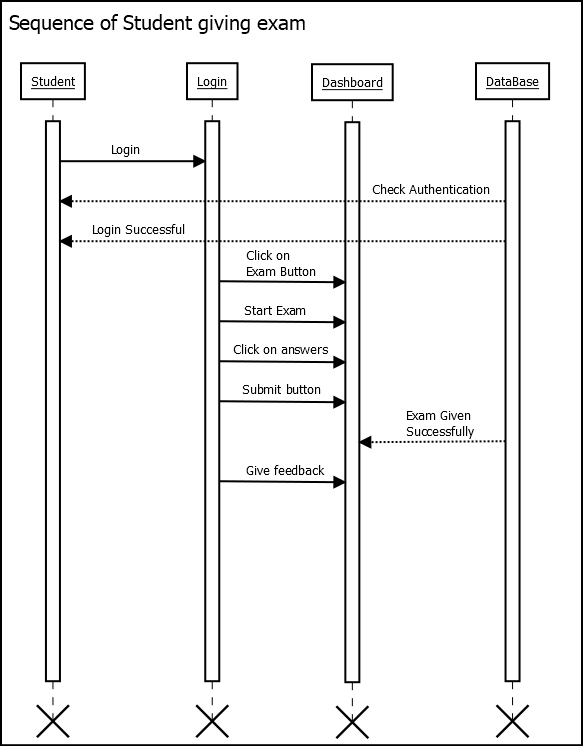
## Class Diagram



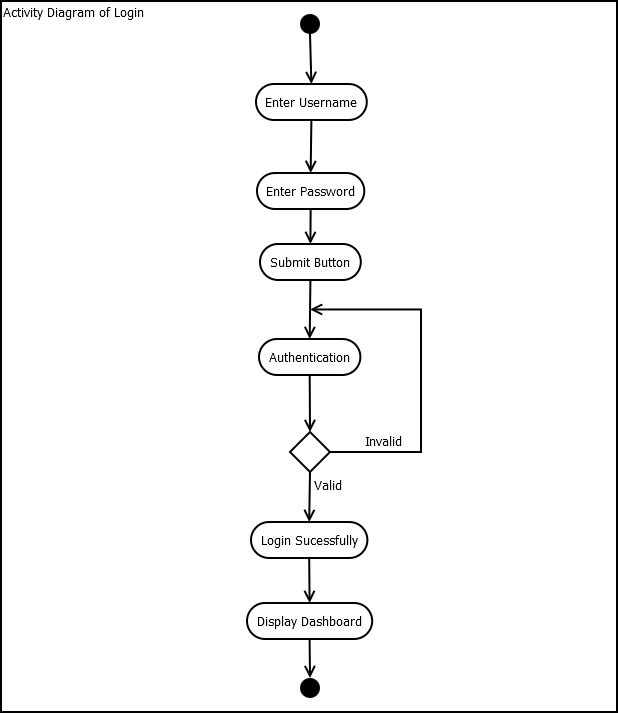
## Interaction Diagram

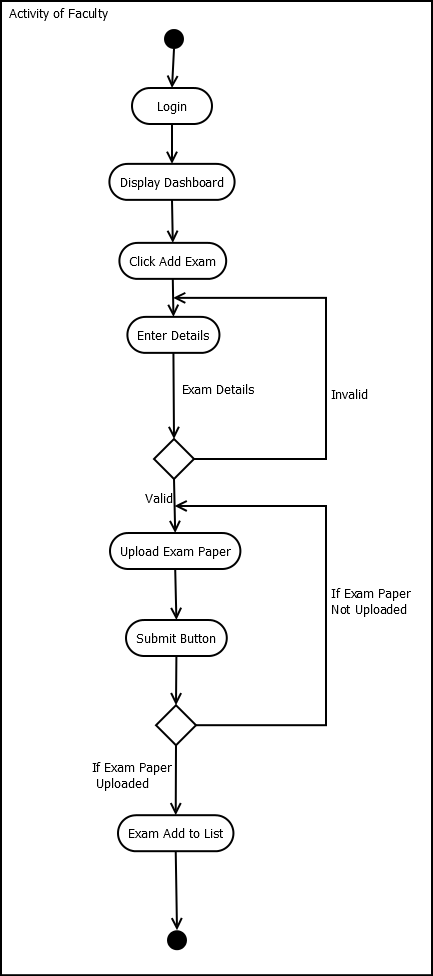


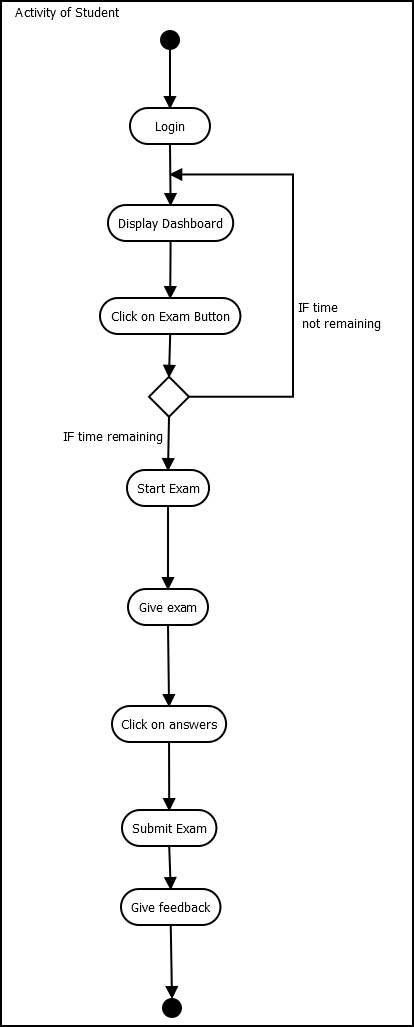




## Activity Diagram







## Data Dictionary

**Table Name:** Faculty

**Table Description:** Stores the basic details FacultyID, Faculty Name, Faculty Branch, Faculty Email.

**Table Level Constraints (if any):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type  <with Size> | Constraint | Description | Sample Data |
| Faculty\_id | number(6) | Not Null | Employee number | 7000 |
| name | Varchar(30) | Not Null | Name of the faculty | Sarjil Sheth |
| department | Varchar(30) | Not Null | Number of dept | MCA |
| degisnation | VarChar(10) | Not Null | Designation | Trusty |
| password | VarChar(15) | Not Null | password | Dgdgergd4r4 |
| date created | Date | Not Null | Student date | 2024-02-29 |

**Table Name:** Student

**Table Description:** Stores Details of Student.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type (with Size) | Constraint | Description | Sample Data |
| Student\_ID | number (6) | **Primary Key** | Student Id | 2712 1421 |
| Enrollment number | number(12) | **Not Null** | Identification | 21018501210001 |
| rollnumber | number(6) | **Not Null** | Student roll no | 1 |
| department | VarChar(30) | Not Null | Student Dept | BCA |
| semester | number(2) | Not Null | Student semester | 6 |
| password | Varchar(20) | Not null | Student password | Sfsfsfsff |
| date created | Date | Not Null | Student date | 2024-02-29 |

**Table Name:** Exam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type (with Size) | Constraint | Description | Sample Data |
| Faculty Id | number(6) | **Not Null** | Faculty Id | 786 |
| division | VarChar (10) | **Not Null** | division | D |
| semester | number(6) | **Not Null** | semester | 6 |
| DEPARTMENT | VarChar (10) | **Not Null** | DEPARTMENT | MCA |
| subject | VarChar (10) | **Not Null** | subject | Maths |
| Start time | date | **Not Null** | Start time | 2024-02-04 |
| End time | date | **Not Null** | End time | 2024-02-05 |
| questions | VarChar (10) | **Not Null** | questions | “{xdfcdgdvg,”bfb”}” |
| Passing marks | number(6) | **Not Null** | Passing marks | 200 |
| Date created | Date | Not Null | Student date | 2024-02-29 |

**Table Name:** exam results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type (with Size) | Constraint | Description | Sample Data |
| Student ID | number(6) | **Foreign Key**  **(Student)** | Student Id | 2712 1421 |
| Exam Id | number(30) | Not Null | Exam id | 565nhtgdgdg |
| Total questions | number(6) | Not Null | Number of questions | 3 |
| Total Marks | number(4) | Not Null | Total Marks | 40 |
| Correct answers | Number(5) | Not Null | Correct answers | 3 |
| None answers | Number(5) | Not Null | None answers | 0 |
| percentage | Number(5) | Not Null | percentage | 100 |
| grade | VarChar(30) | Not Null | Student grade | A |
| Exam duration | VarChar(30) | Not Null | Exam duration | 171:-10 |

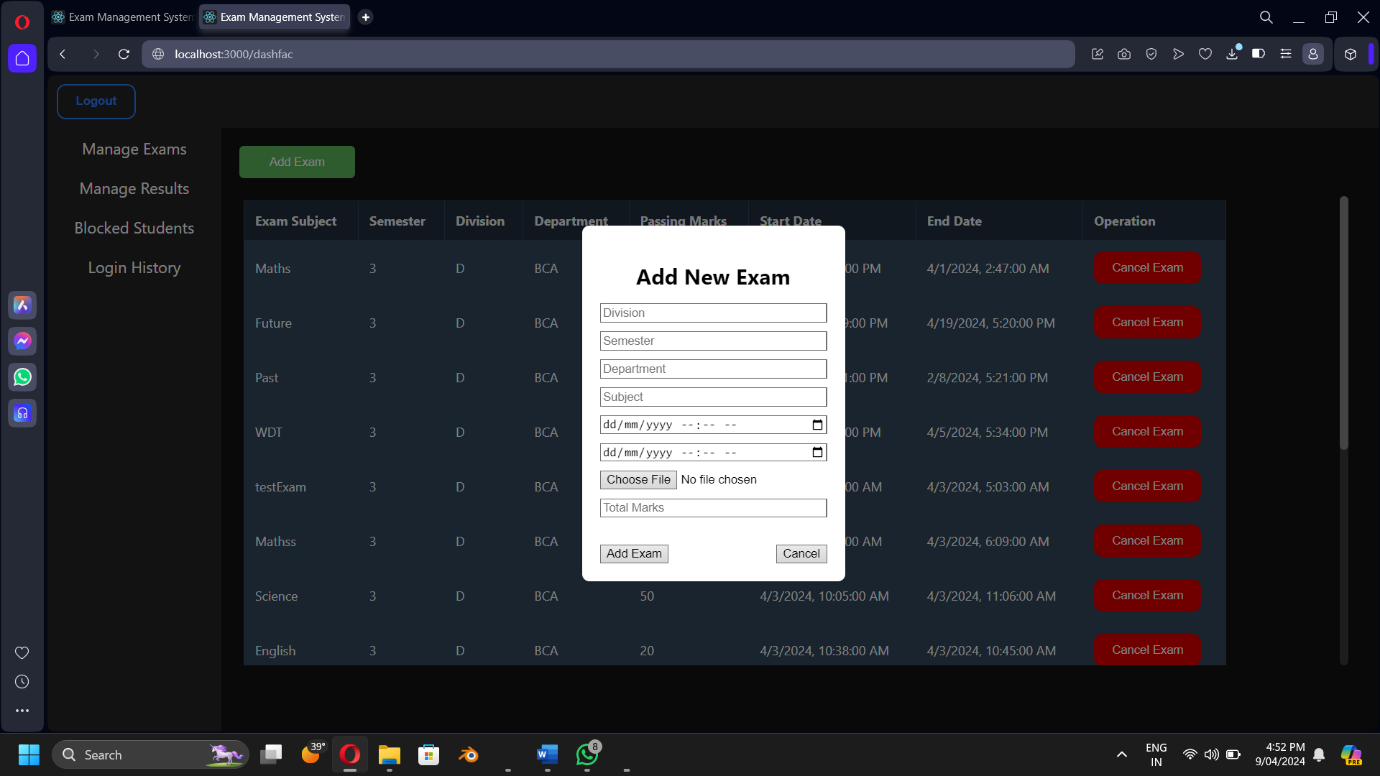
**Table Name:** Login history

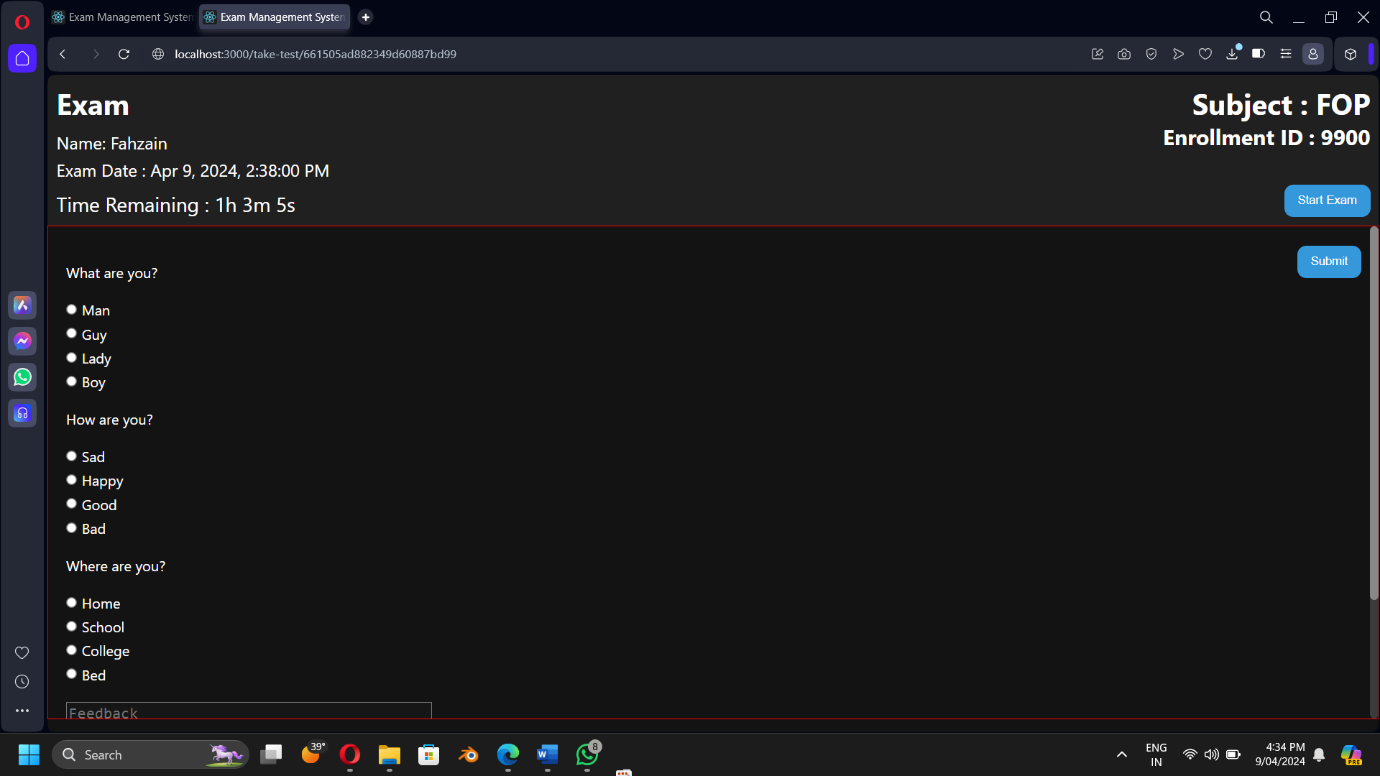
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type (with Size) | Constraint | Description | Sample Data |
| position | VarChar (10) | **Not Null** | position | Student |
| User Name | number(16) | **Not Null** | User number | 89789 |
| Login time | Date | Not Null | login date | 2024-02-29 |
| task | VarChar (10) | **Not Null** | task | login |

# Development

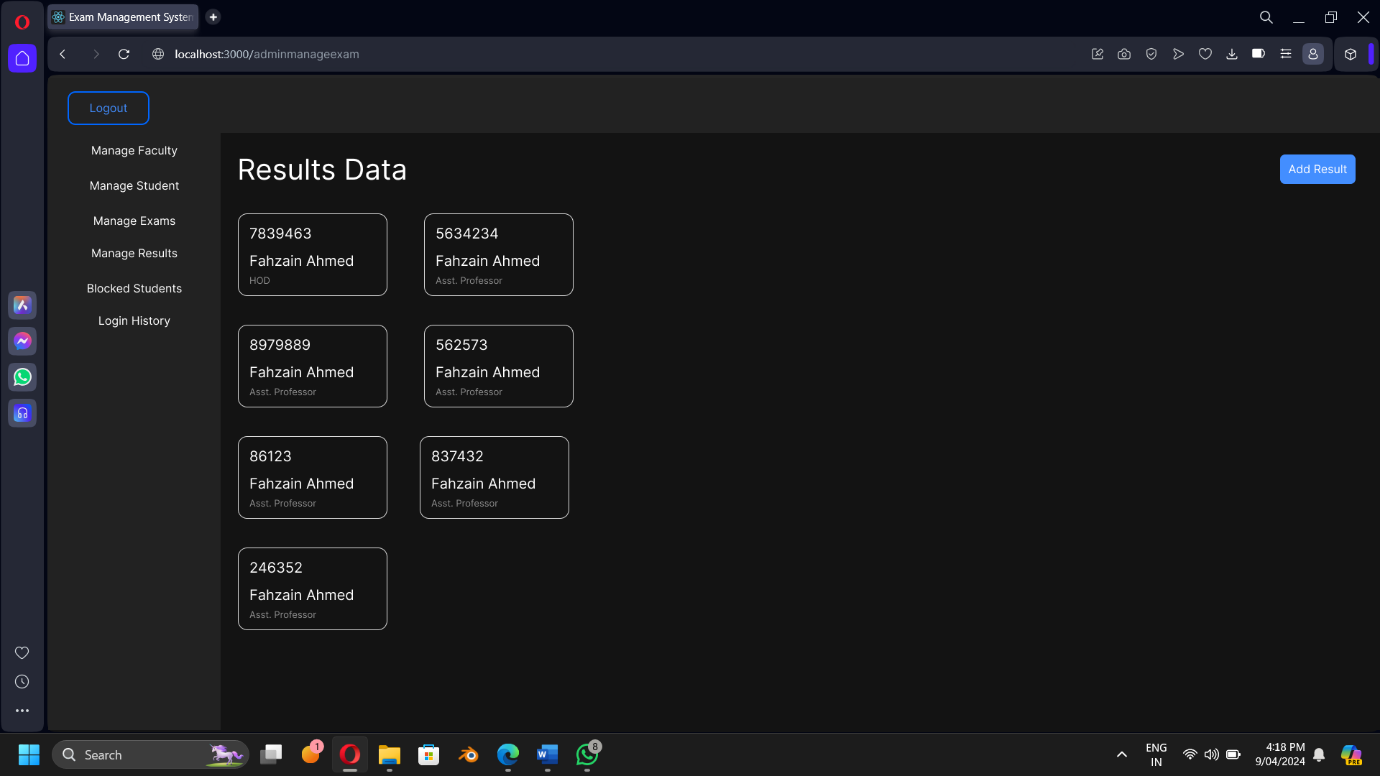
***In our project, we have three members assigned to different coding tasks. One member is responsible for UI design. Once the UI design is completed, we've begun developing the main code to connect to the database. The remaining two members are working on both frontend and backend tasks. We've divided our responsibilities; I've focused on creating the faculty dashboard and its operations, while the third member has concentrated on the admin and student dashboards. We're collaborating using Git Desktop to share our code and work together towards achieving project success.***

***Faculty Adds Exam***

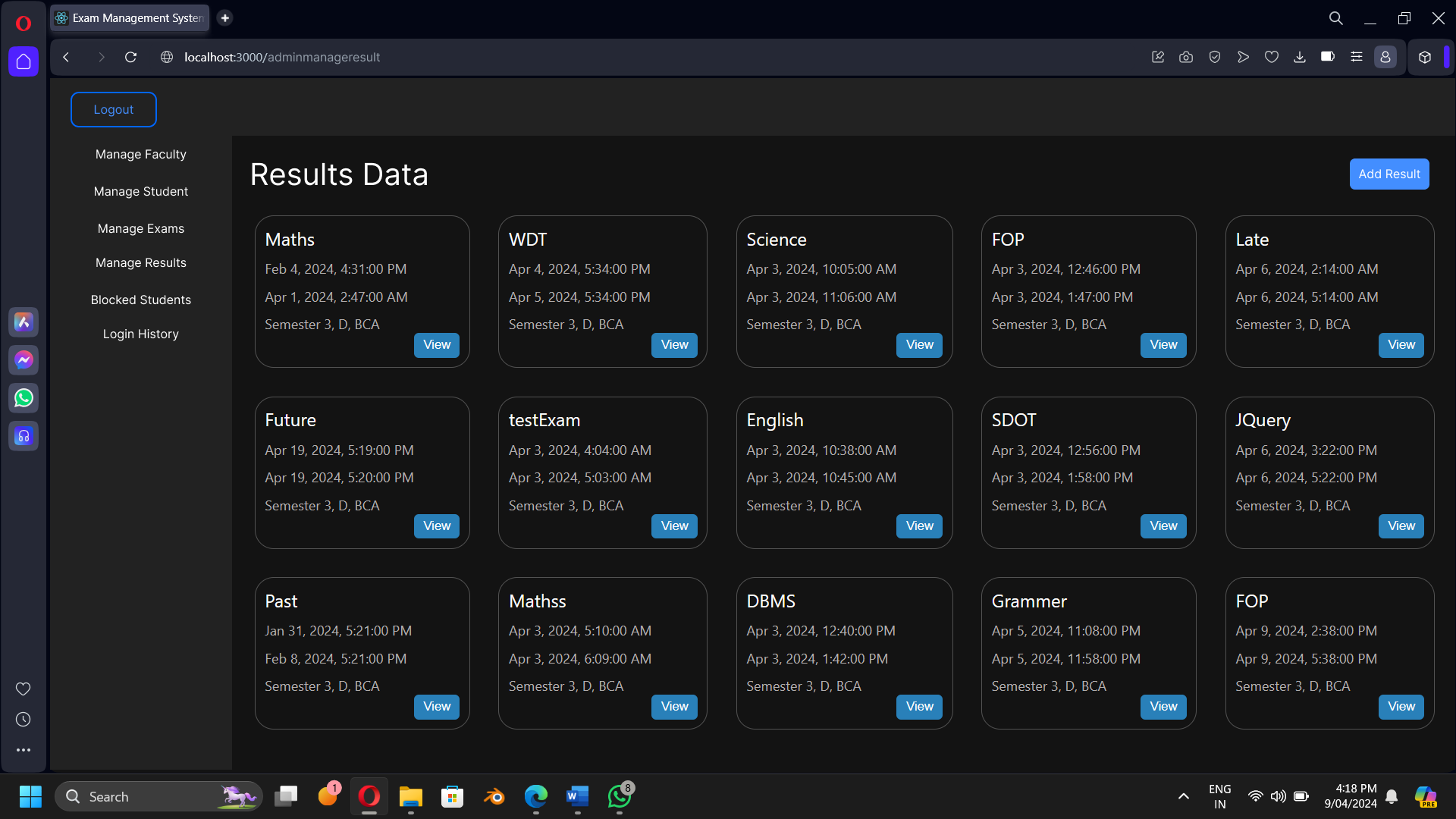


***Student Gives Exam***

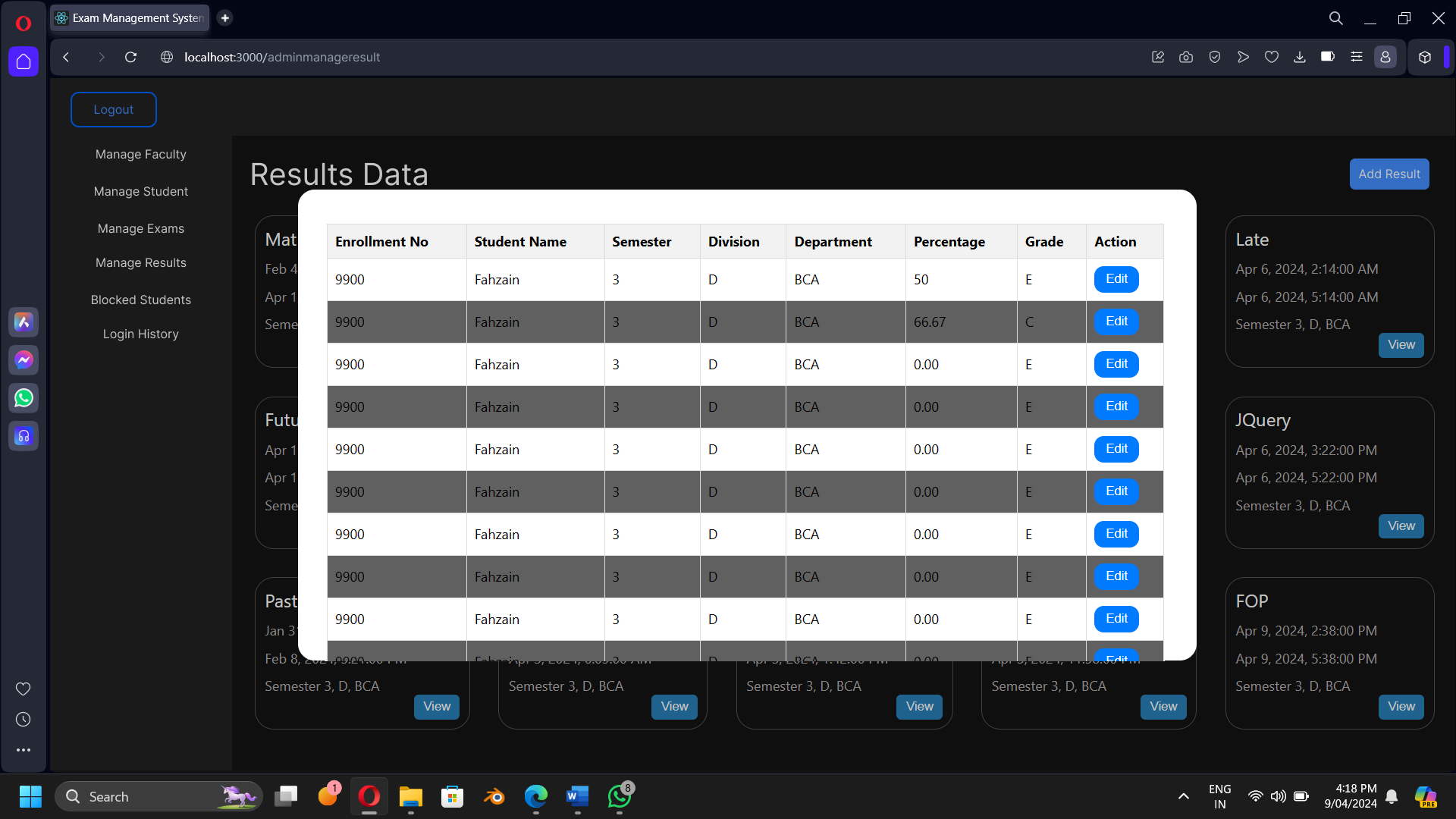
***Admin Manage Result – Choose Faculty***



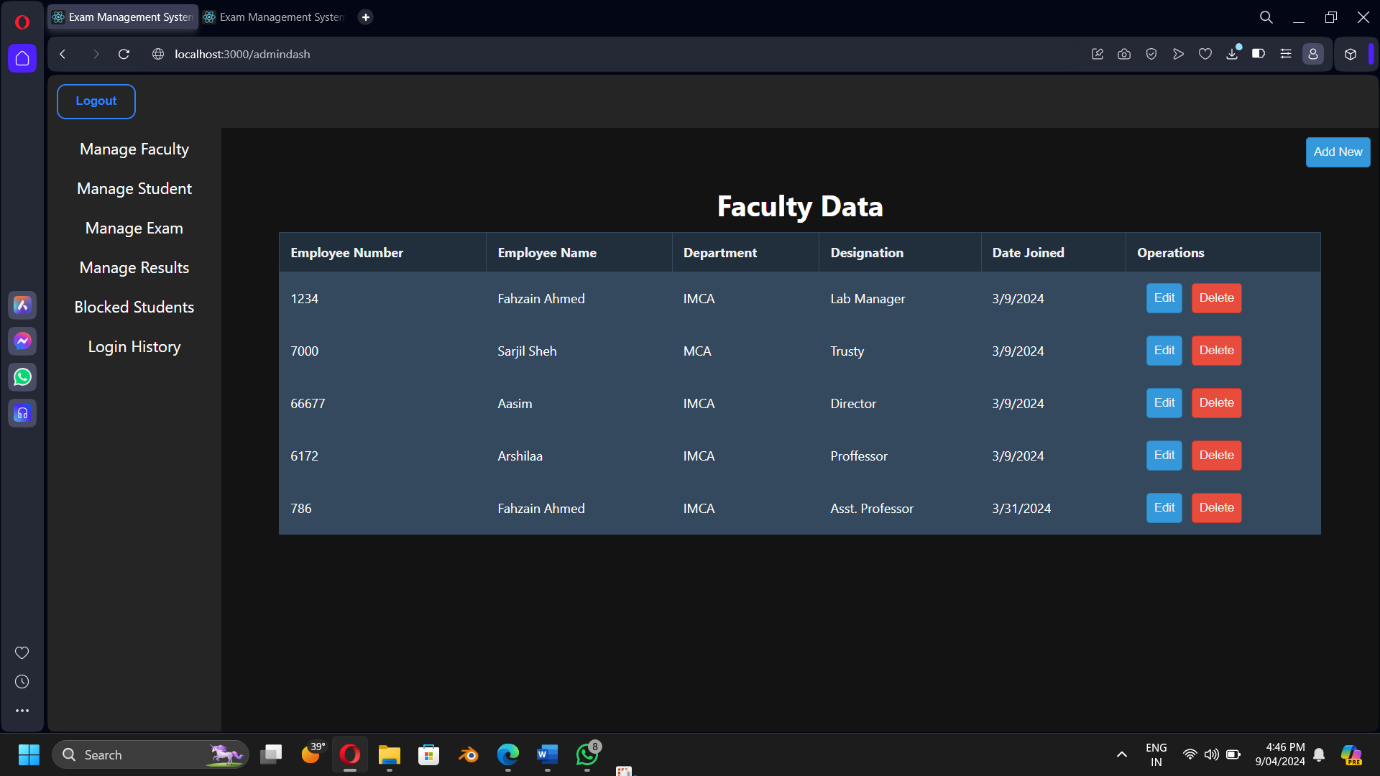
***Admin Manage Result – Choose Exam***

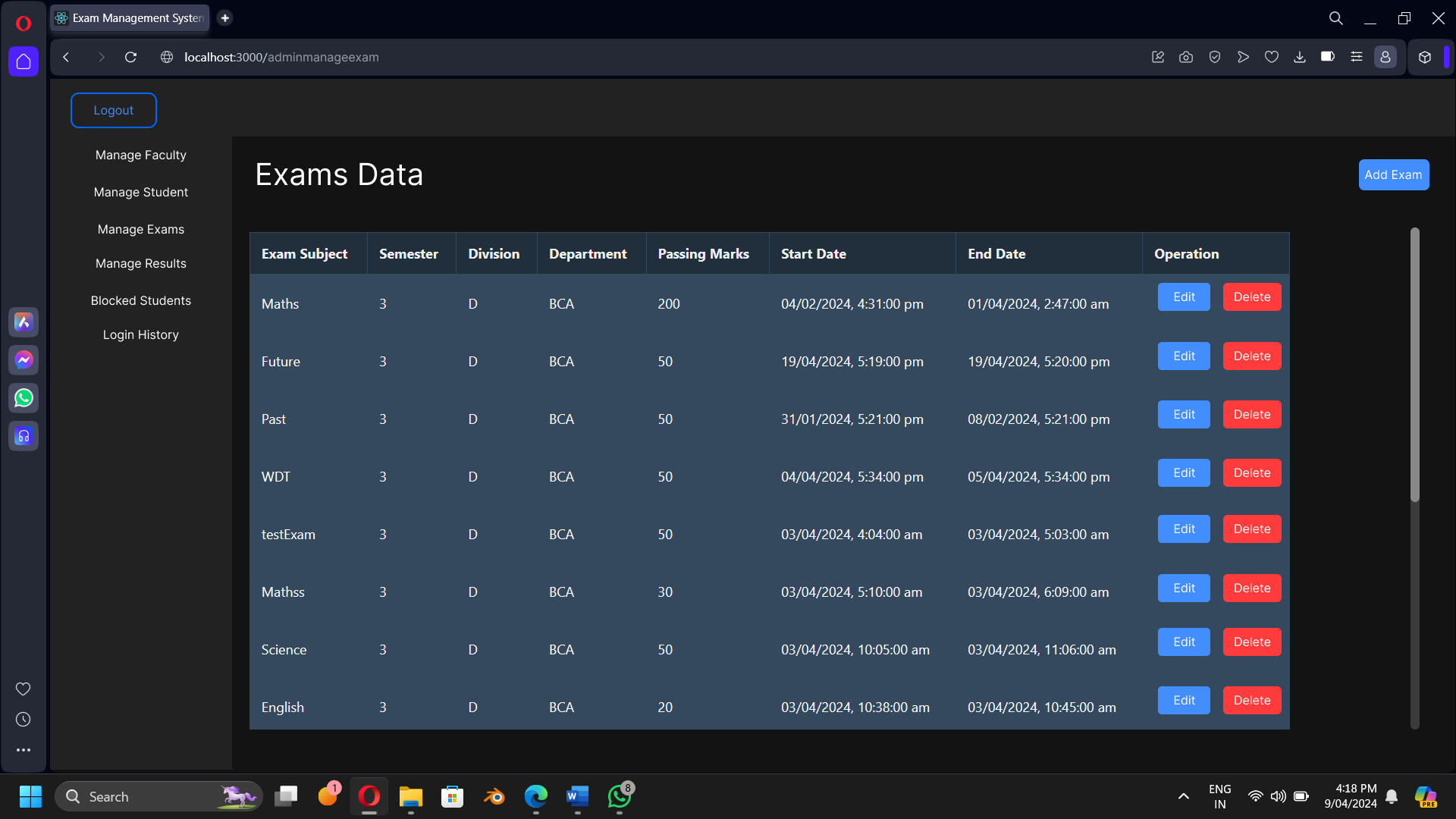
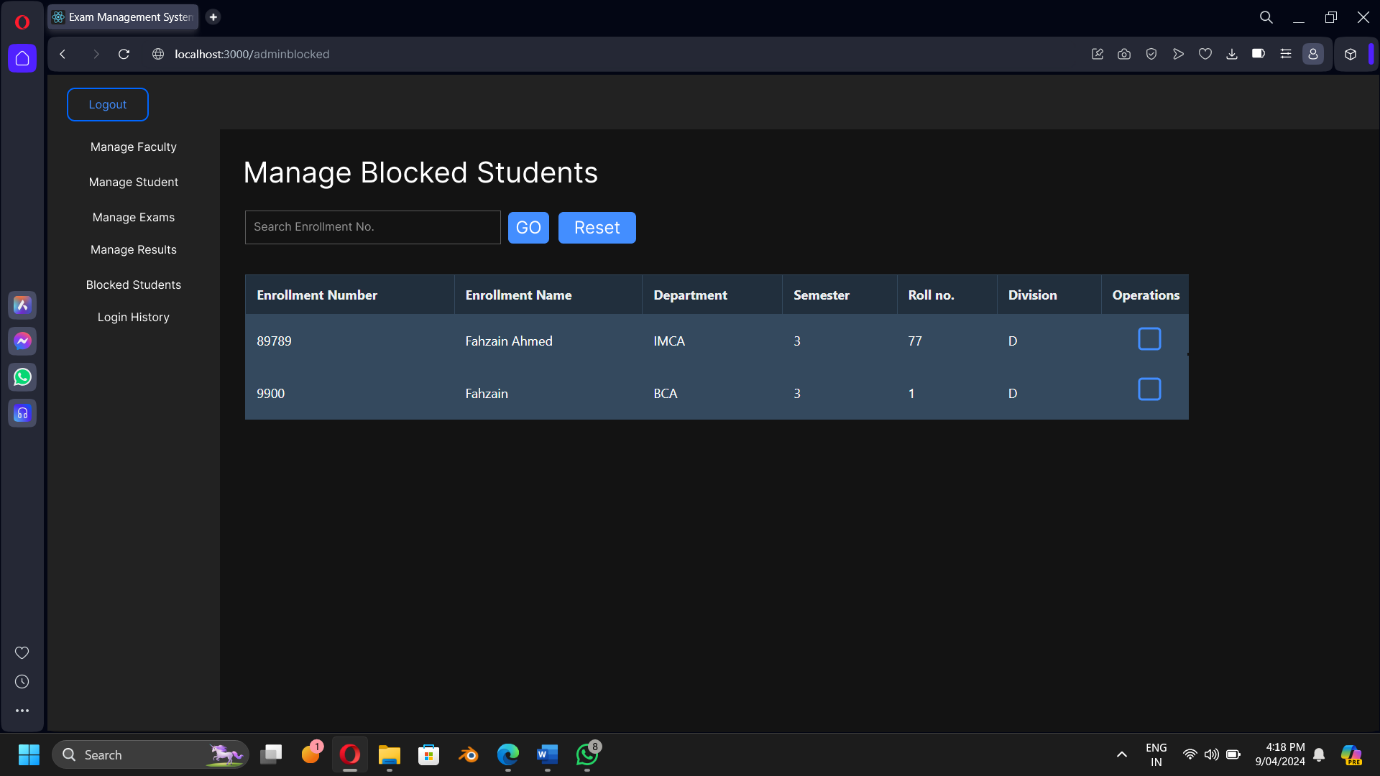


***Admin Manage Result – Choose Student***

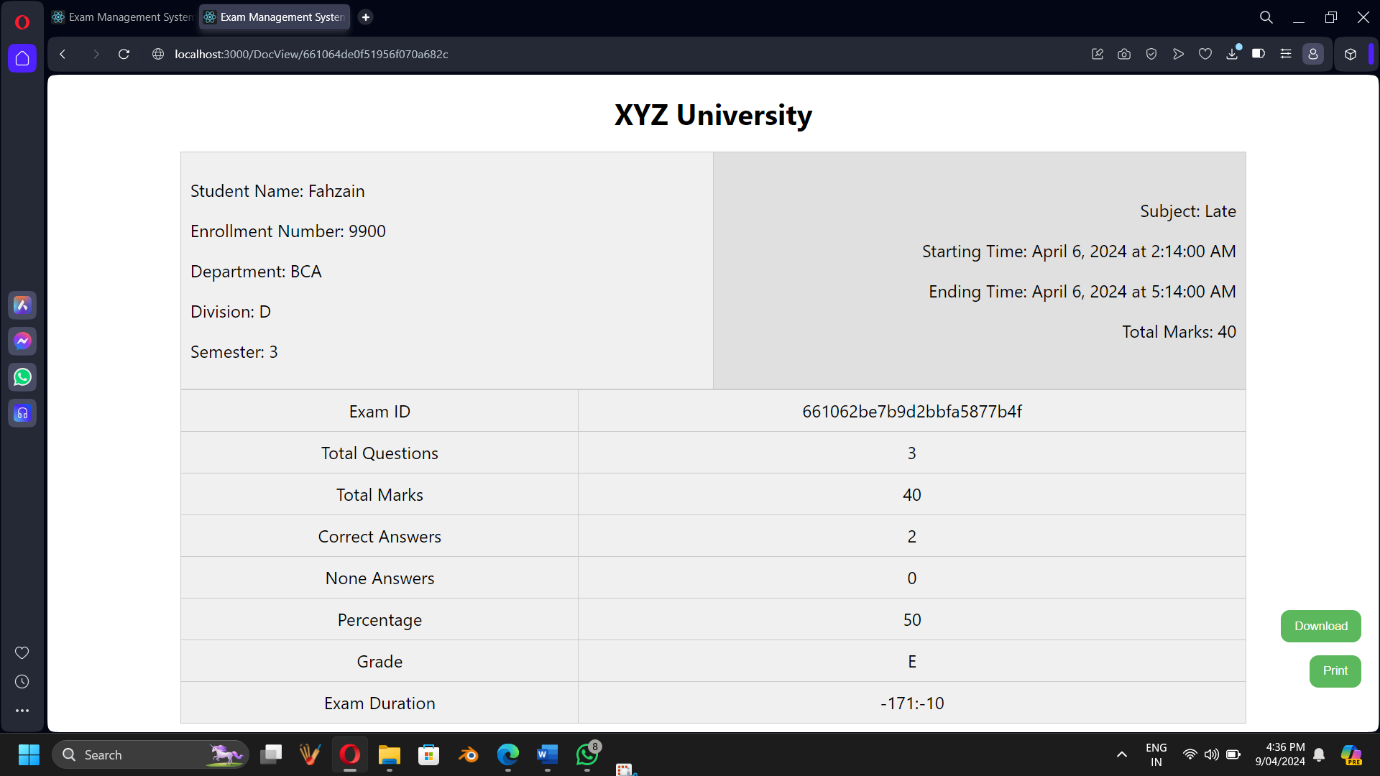


***Admin Manage Faculty***

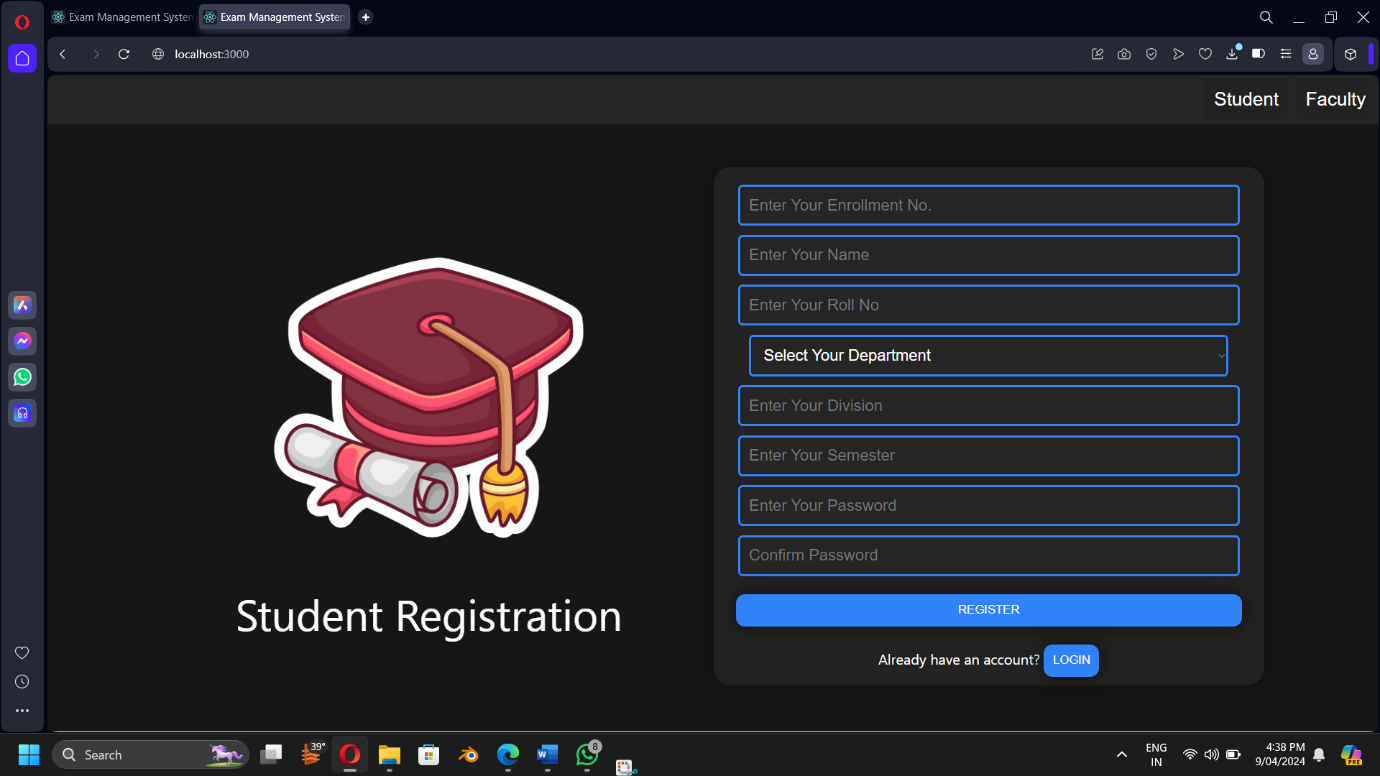
 ***Admin Manage Result – Choose Exam***

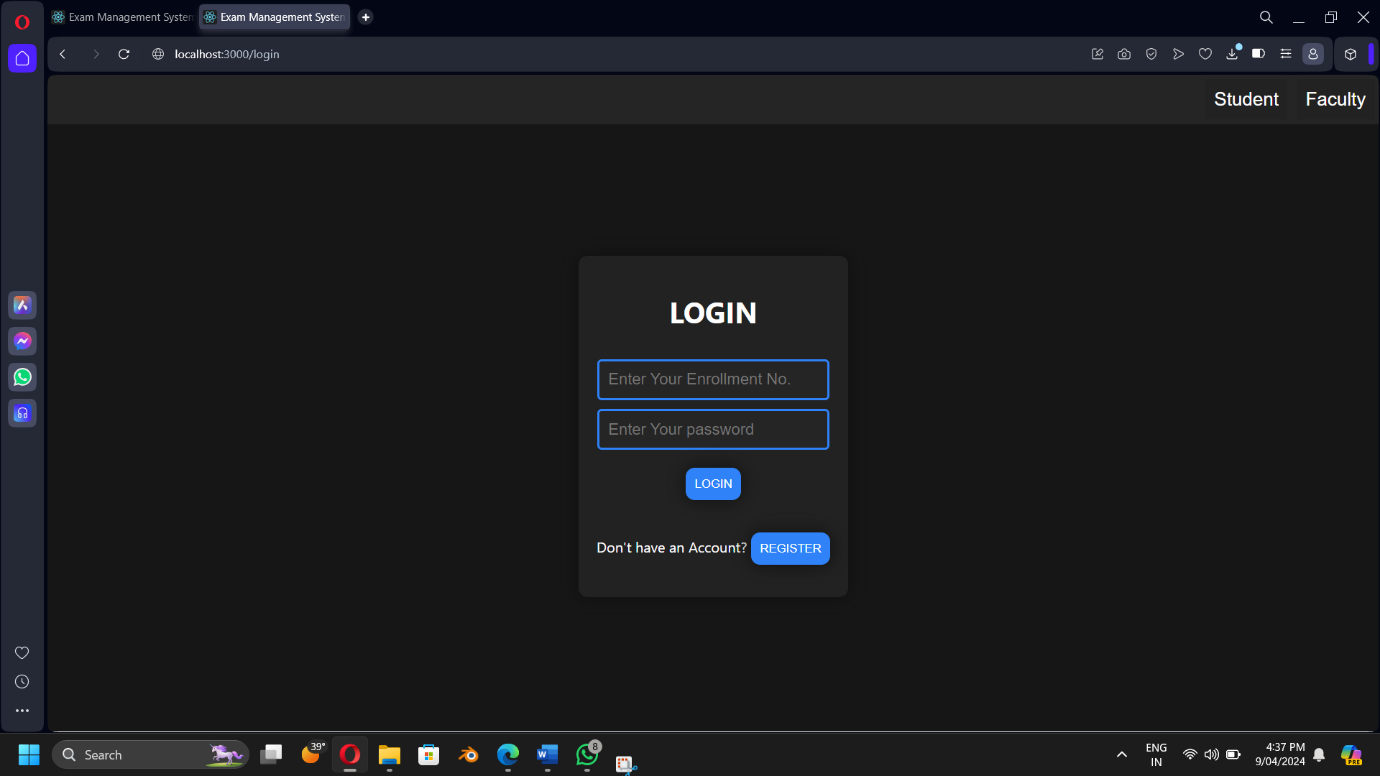
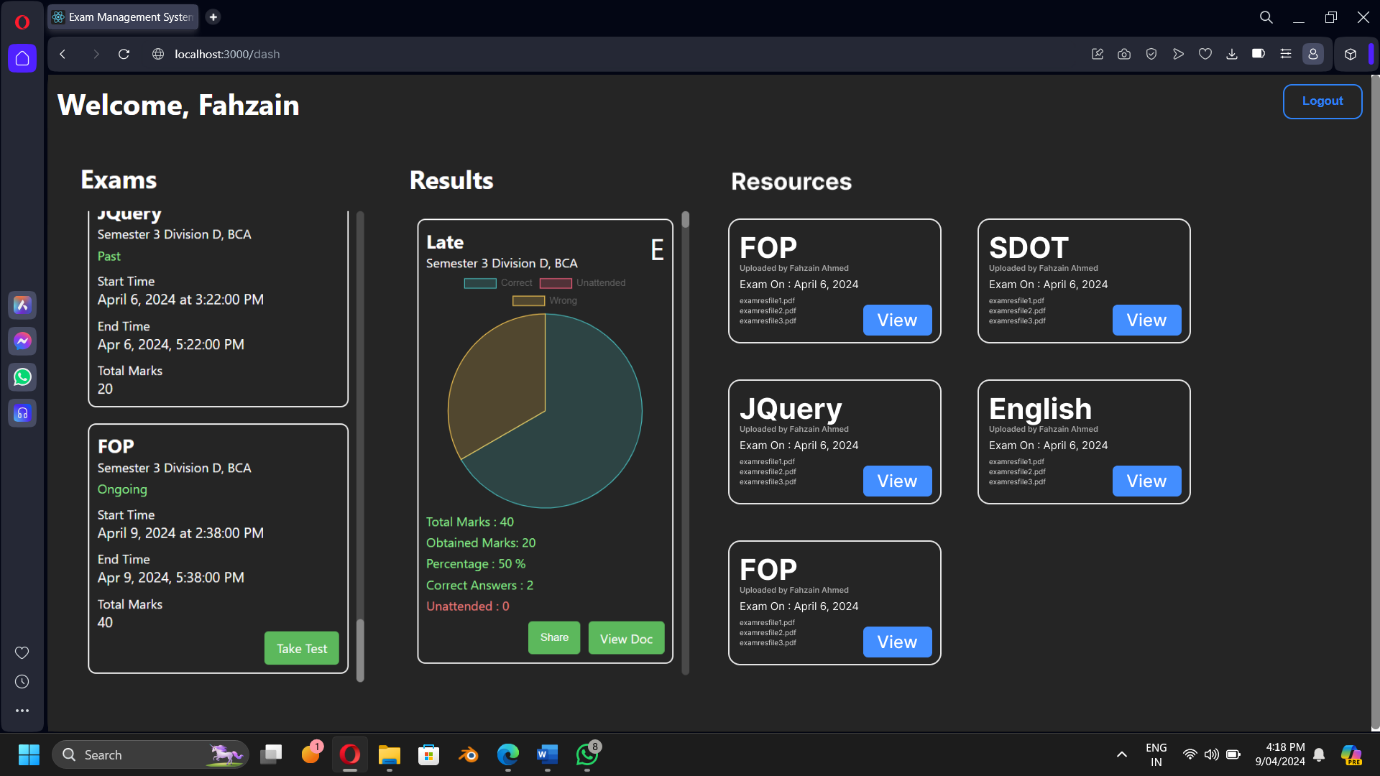
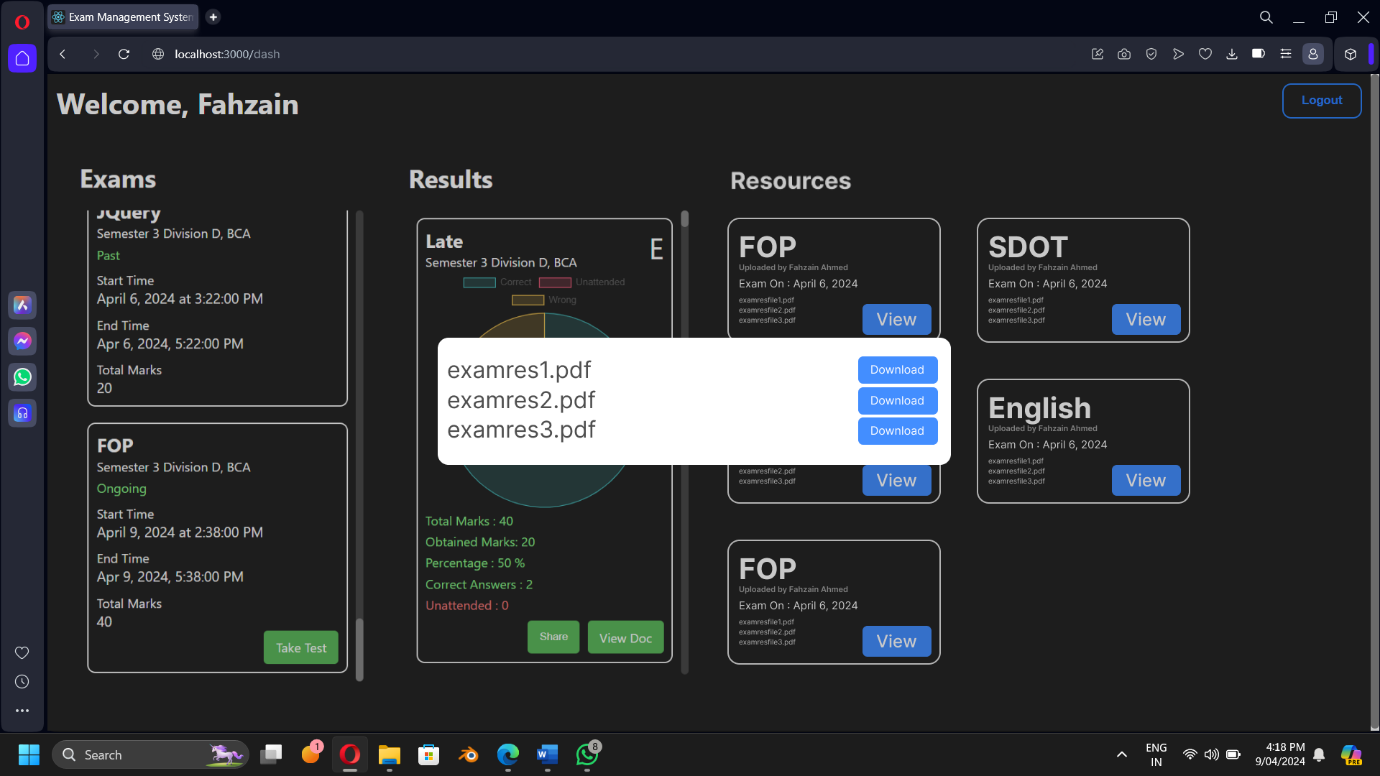
***Admin Manage Exam******Admin Manage Blocked Students***

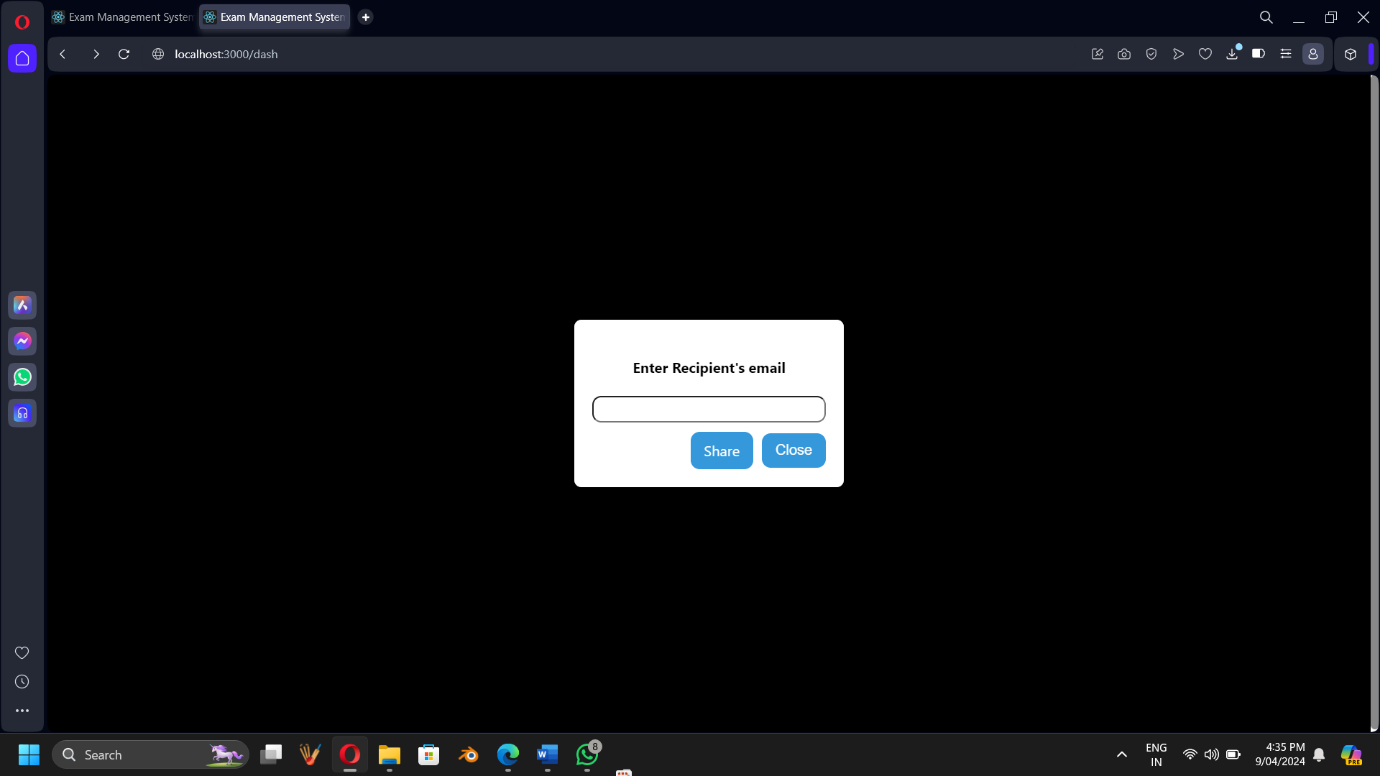
***Student View Result***

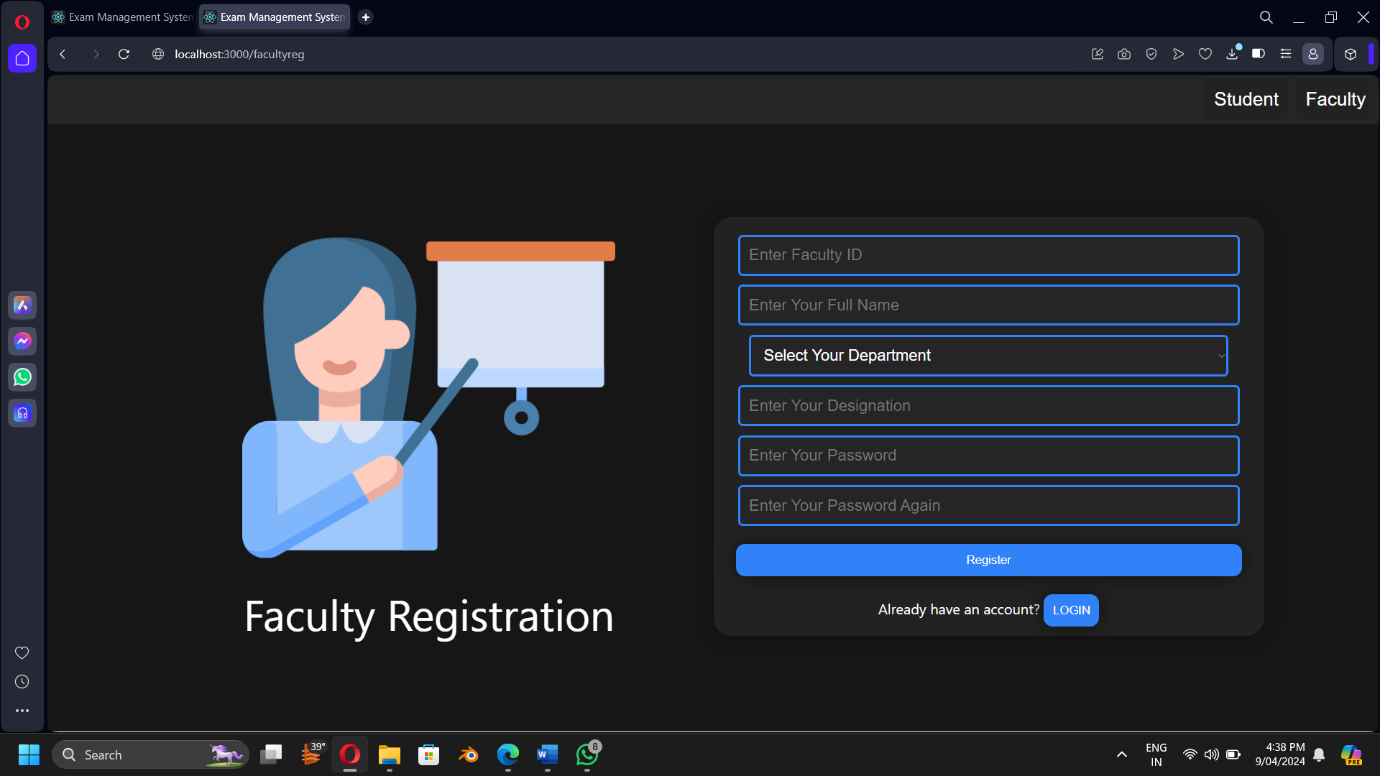


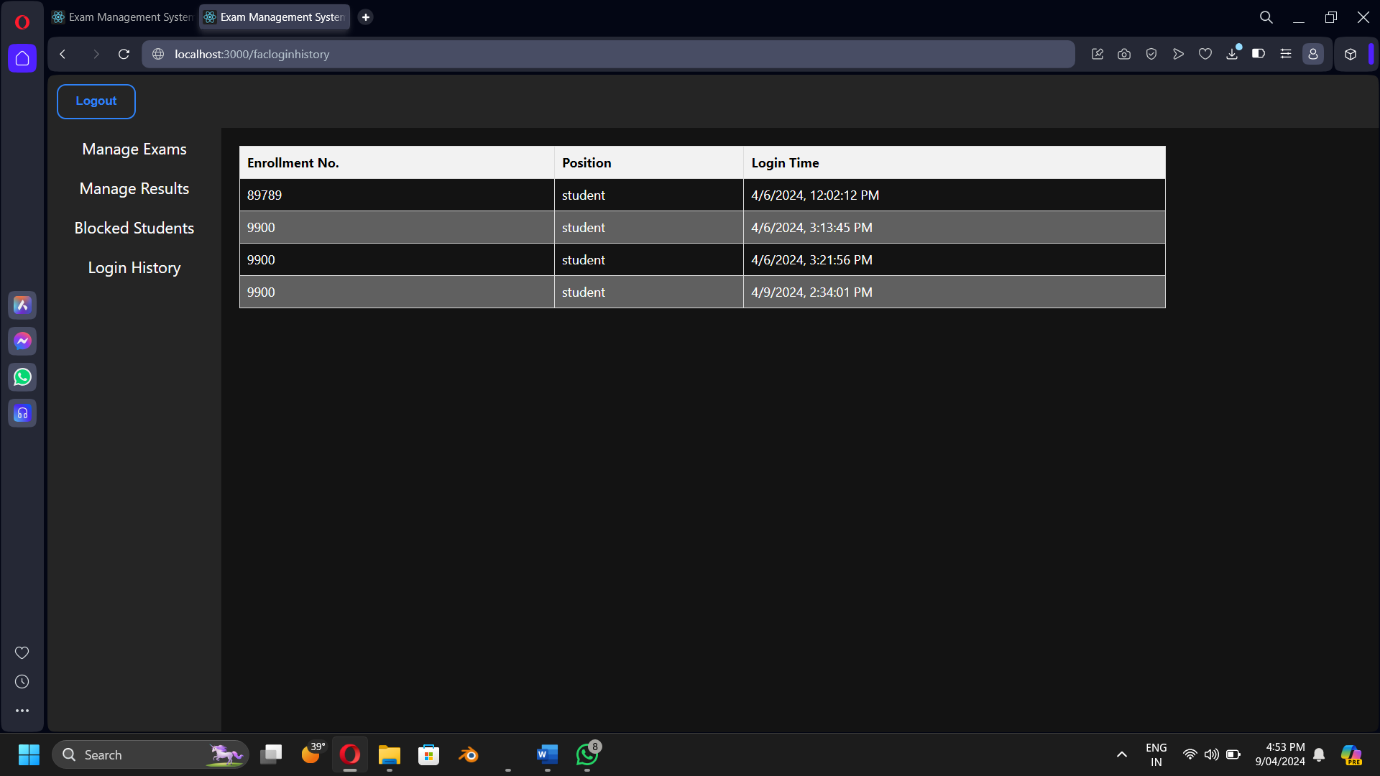
***Student Registration***

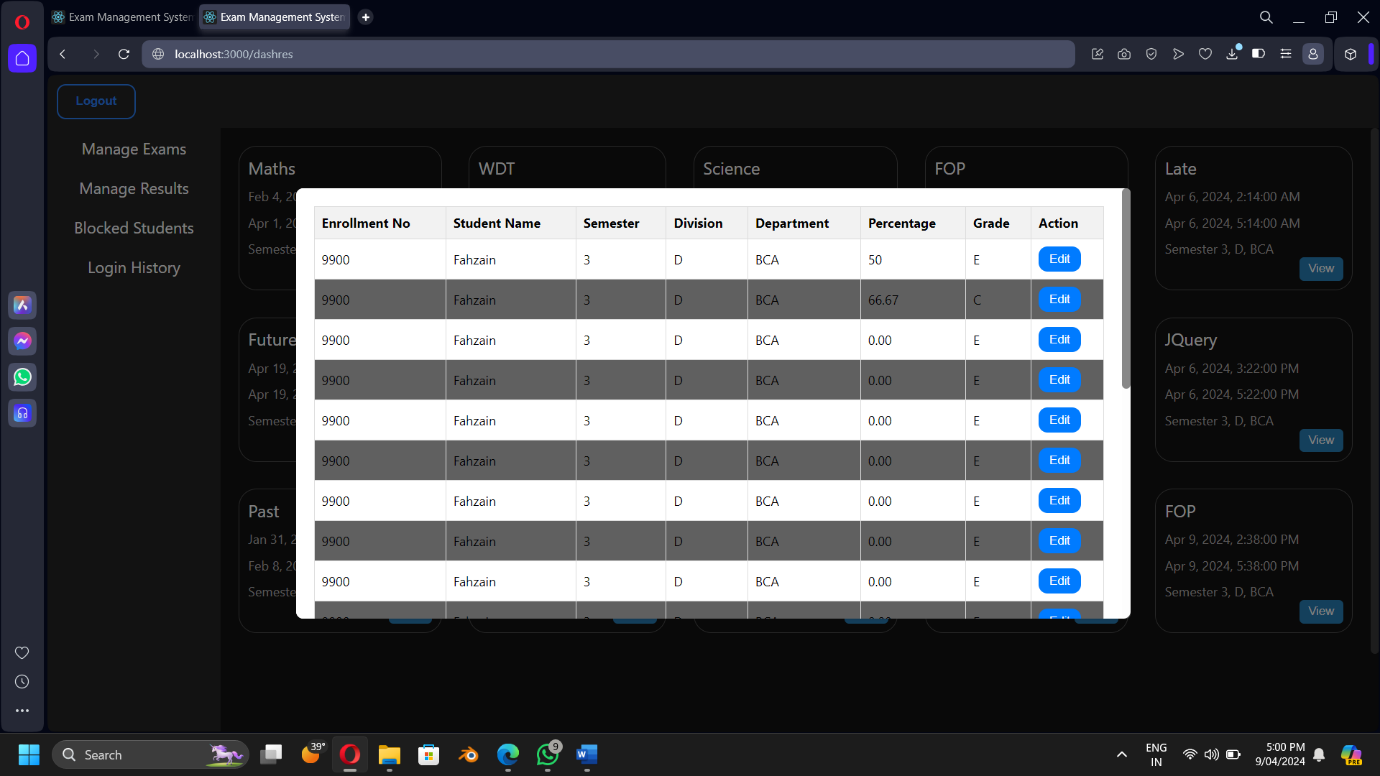


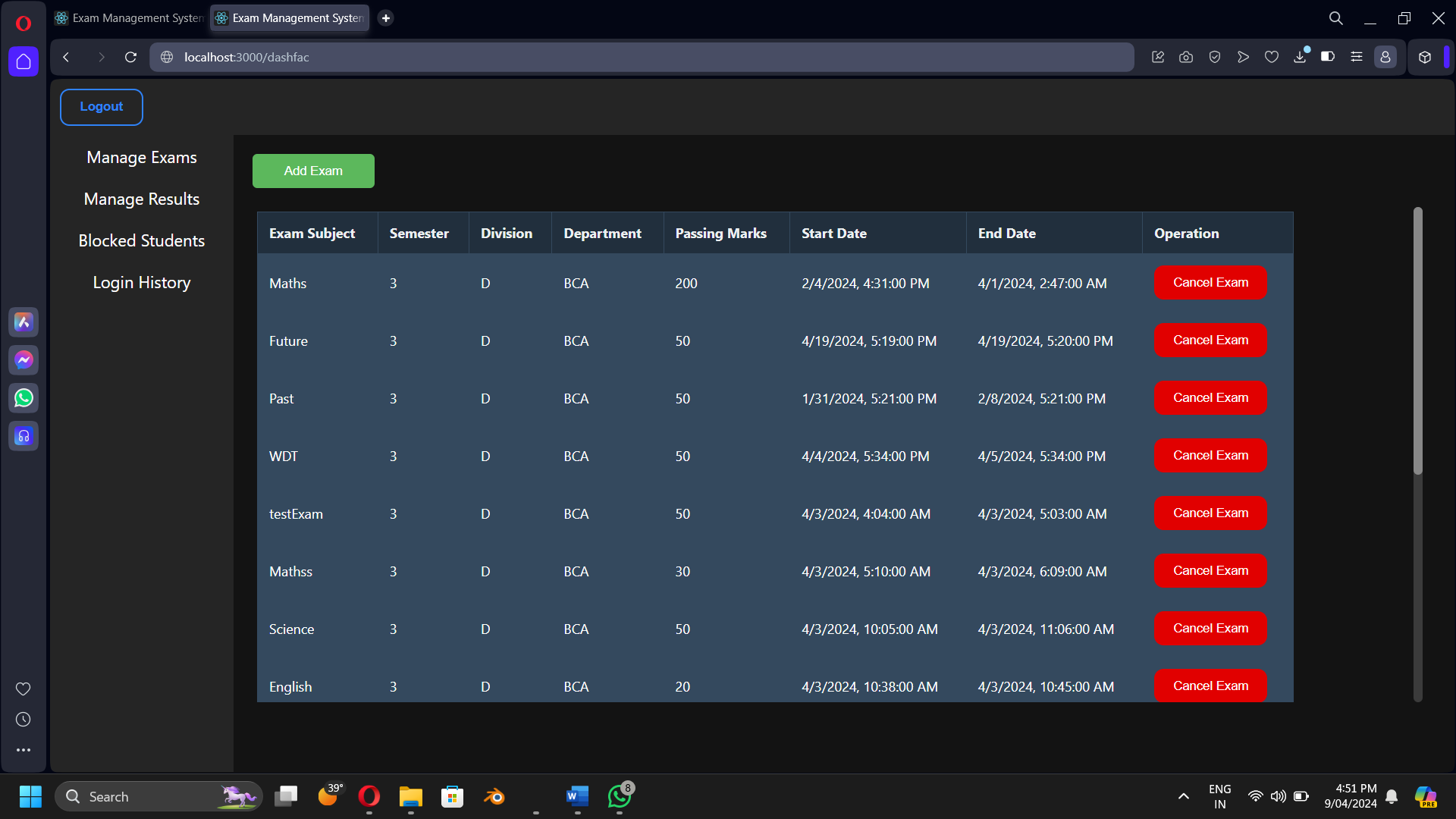
***Student Login******Student Dashboard******Student Dashboard – View Resources***

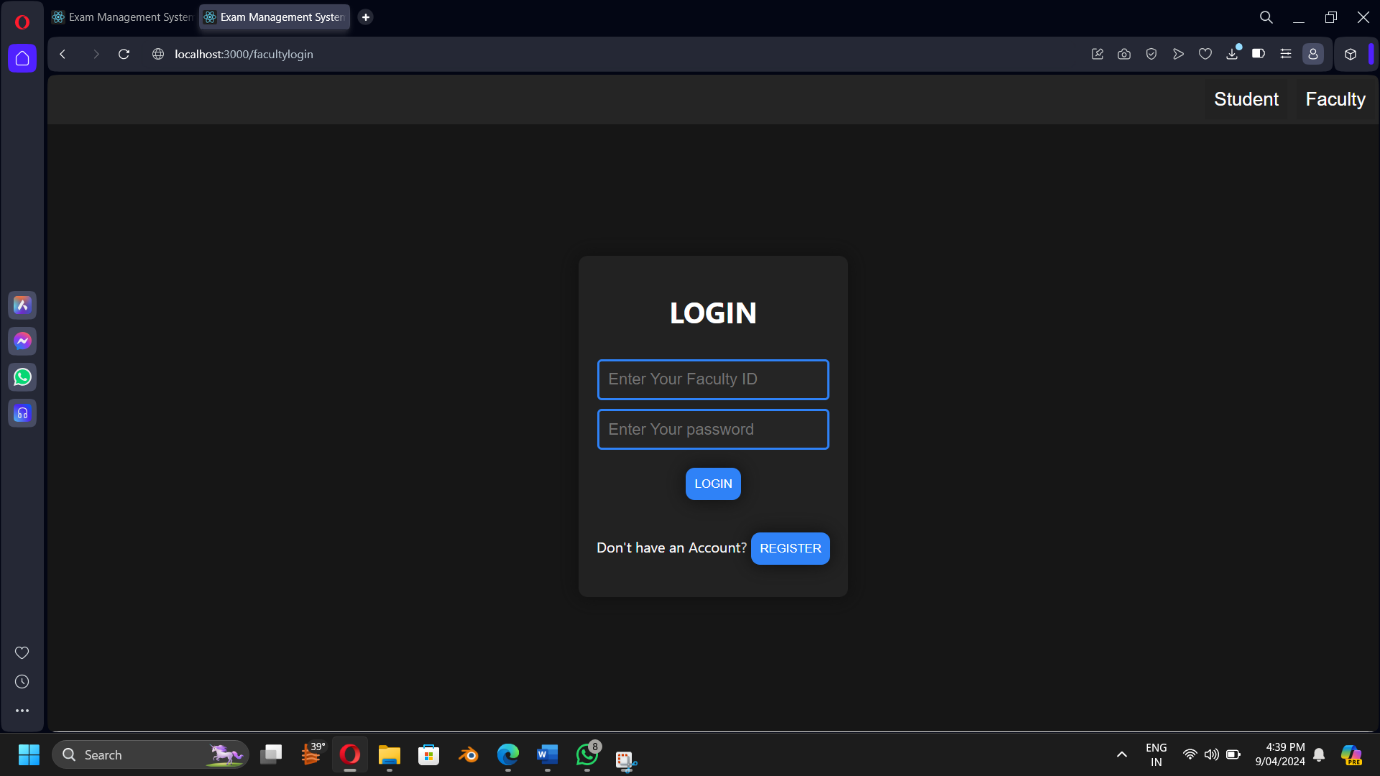
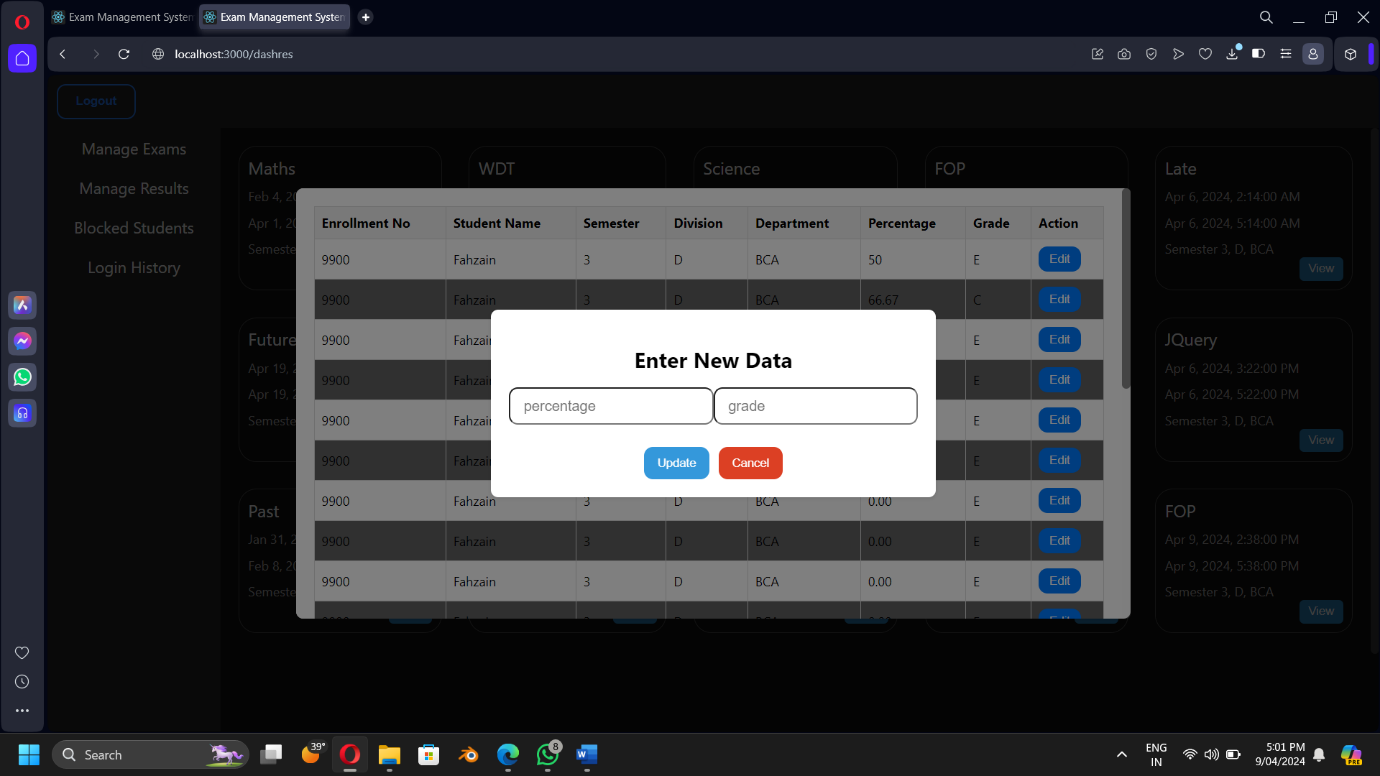
***Student Share Result***

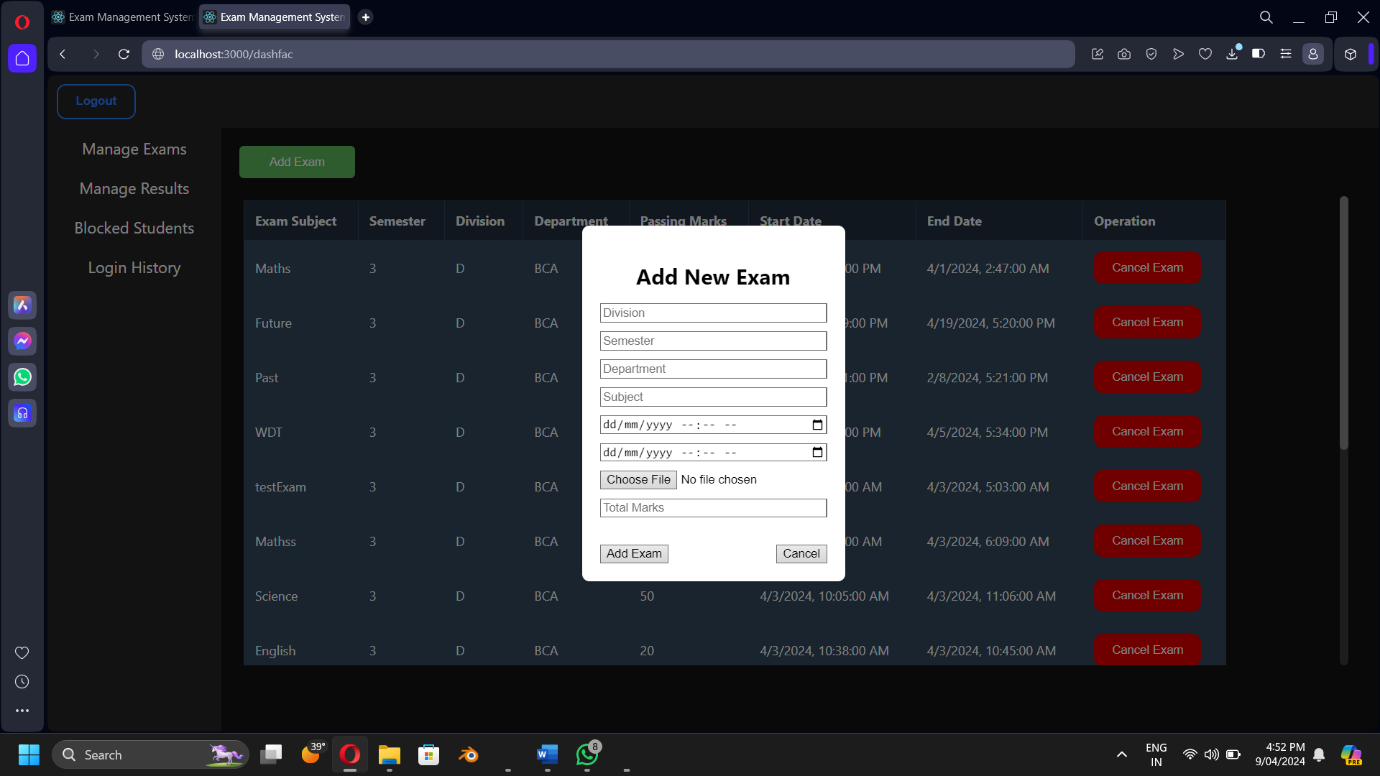
***Faculty Registration***

***Faculty Login History***

***Faculty Manage Results***

***Faculty Manage Exams***

***Faculty Login*** ***Faculty Edit Result***

***Faculty Add Exams***

## Coding Standards

***During development, we stick to clear rules for writing code:***

***1.Naming: We give things clear names so others can understand what they do.***

***2. Organization: We split our code into small parts that each do one thing well.***

***3. Comments: We add notes to explain tricky parts of our code.***

***4. Testing: We check our code to make sure it works like it should.***

***5. Git: We use Git to save and share our code, so everyone stays on the same page.***

***By following these rules, our code is easier to read, understand, and work with.***

# Agile Documentation

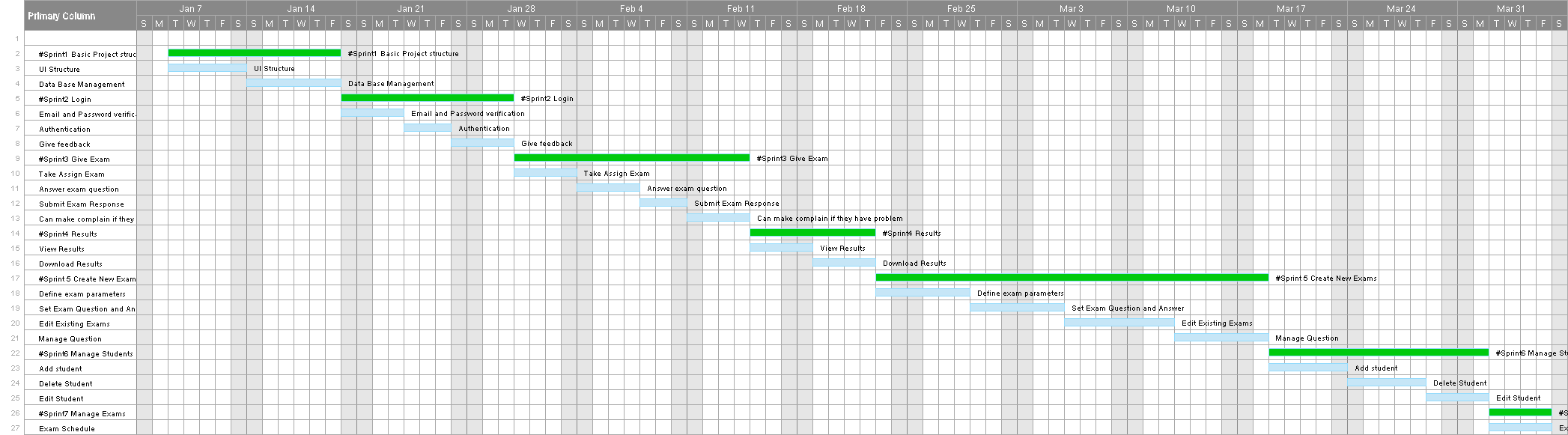
## Agile Project Charter

|  |  |
| --- | --- |
| General Project Information | |
| Project Name | **Exam Management System** |
| Project Champion | **Ahmed Fahzain, Aasim Saiyed, Sheth Mahamad Sarjil** |
| Project Sponsor | **Not applicable (N/A)** |
| Project Manager | **Professor Sangeeta Rajole** |
| Stakeholders | **L.J Institute Of Computer Application** |
| Expected Start Date | **26-December-2023** |
| Expected Completion Date | **6-April-2024** |

|  |  |
| --- | --- |
| Project Details | |
| Mission | **To streamline exam management processes efficiently.** |
| Vision | **To create a user-friendly platform for seamless exam administration** |
| Scope | **Developing a comprehensive system for exam scheduling, registration, and result management.** |

## 

## Agile Road Map



## Agile Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Name | Duration | Start | Finish | Status |
| #Sprint1 Basic Project structure | 11d | 01-09-2024 | 01-19-2024 | Complete |
| UI Structure | 5d | 01-09-2024 | 01-13-2024 | Complete |
| Data Base Management | 6d | 01-14-2024 | 01-19-2024 | Complete |
| #Sprint2 Login | 11d | 01-20-2024 | 01-30-2024 | Complete |
| Email and Password verification | 4d | 01-20-2024 | 01-23-2024 | Complete |
| Authentication | 3d | 01-24-2024 | 01-26-2024 | Complete |
| Give feedback | 4d | 01-27-2024 | 01-30-2024 | Complete |
| #Sprint3 Give Exam | 15d | 01-31-2024 | 02-14-2024 | Complete |
| Take Assign Exam | 4d | 01-31-2024 | 02-03-2024 | Complete |
| Answer exam question | 4d | 02-04-2024 | 02-07-2024 | Complete |
| Submit Exam Response | 3d | 02-08-2024 | 02-10-2024 | Complete |
| Can make complain if they have problem | 4d | 02-11-2024 | 02-14-2024 | Complete |
| #Sprint4 Results | 8d | 02-15-2024 | 02-22-2024 | Complete |
| View Results | 4d | 02-15-2024 | 02-18-2024 | Complete |
| Download Results | 4d | 02-19-2024 | 02-22-2024 | Complete |
| #Sprint 5 Create New Exams | 25d | 02-23-2024 | 03-18-2024 | Complete |
| Define exam parameters | 6d | 02-23-2024 | 02-28-2024 | Complete |
| Set Exam Question and Answer | 6d | 02-29-2024 | 03-05-2024 | Complete |
| Edit Existing Exams | 7d | 03-06-2024 | 03-12-2024 | Complete |
| Manage Question | 6d | 03-13-2024 | 03-18-2024 | Complete |
| #Sprint6 Manage Students | 14d | 03-19-2024 | 04-01-2024 | Complete |
| Add student | 5d | 03-19-2024 | 03-23-2024 | Complete |
| Delete Student | 5d | 03-24-2024 | 03-28-2024 | Complete |
| Edit Student | 4d | 03-29-2024 | 04-01-2024 | Complete |
| #Sprint7 Manage Exams | 4d | 04-02-2024 | 04-05-2024 | Complete |
| Exam Schedule | 4d | 04-02-2024 | 04-05-2024 | Complete |

## Agile User Story ( Minimum 3 Tasks)

|  |  |  |  |
| --- | --- | --- | --- |
| User Story ID | As s(type of user) | I want to perform | So that I can(achieve some task) |
| 1 | Admin | Start exams | Student exams will be start |
| 2 | Admin | Cancel exams | Student exams must be cancel |
| 3 | Admin | Make schedule | Make schedule for exams |
| 4 | Admin | Manage Results | Manage Results of students |
| 5 | Admin | Manage Students | Manage students add, them remove them and also block them |
| 6 | Admin | Manage Complains and feedback | Manage complain and solve them and Manage feedback and review them |
| 7 | Faculty | Start exams | Student exams will be start |
| 8 | Faculty | Make schedule | Make schedule for exams |
| 9 | Faculty | Cancel exams | Student exams must be cancel |
| 10 | Faculty | Upload Paper | Upload exam paper for students |
| 11 | Student | Give exam | Gives their exam |
| 12 | Student | View schedule | View schedule for exam |
| 13 | Student | View Results and Download | View their marks and download their marks |
| 14 | Student | Make complain | Make complain if they have issue |
| 15 | Student | Give Feedback | Give feedback about exams |

## Agile Release Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task Name | Duration | Start | Finish | Status | Release Date |
| #Sprint1 Basic Project structure | 11d | 01-09-2024 | 01-19-2024 | Complete | 01-19-2024 |
| UI Structure | 5d | 01-09-2024 | 01-13-2024 | Complete | 01-13-2024 |
| Data Base Management | 6d | 01-14-2024 | 01-19-2024 | Complete | 01-19-2024 |
| #Sprint2 Login | 11d | 01-20-2024 | 01-30-2024 | Complete | 01-30-2024 |
| Email and Password verification | 4d | 01-20-2024 | 01-23-2024 | Complete | 01-23-2024 |
| Authentication | 3d | 01-24-2024 | 01-26-2024 | Complete | 01-26-2024 |
| Give feedback | 4d | 01-27-2024 | 01-30-2024 | Complete | 01-30-2024 |
| #Sprint3 Give Exam | 15d | 01-31-2024 | 02-14-2024 | Complete | 02-14-2024 |
| Take Assign Exam | 4d | 01-31-2024 | 02-03-2024 | Complete | 02-03-2024 |
| Answer exam question | 4d | 02-04-2024 | 02-07-2024 | Complete | 02-07-2024 |
| Submit Exam Response | 3d | 02-08-2024 | 02-10-2024 | Complete | 02-10-2024 |
| Can make complain if they have problem | 4d | 02-11-2024 | 02-14-2024 | Complete | 02-14-2024 |
| #Sprint4 Results | 8d | 02-15-2024 | 02-22-2024 | Complete | 02-22-2024 |
| View Results | 4d | 02-15-2024 | 02-18-2024 | Complete | 02-18-2024 |
| Download Results | 4d | 02-19-2024 | 02-22-2024 | Complete | 02-22-2024 |
| #Sprint 5 Create New Exams | 25d | 02-23-2024 | 03-18-2024 | Complete | 03-18-2024 |
| Define exam parameters | 6d | 02-23-2024 | 02-28-2024 | Complete | 02-28-2024 |
| Set Exam Question and Answer | 6d | 02-29-2024 | 03-05-2024 | Complete | 03-05-2024 |
| Edit Existing Exams | 7d | 03-06-2024 | 03-12-2024 | Complete | 03-12-2024 |
| Manage Question | 6d | 03-13-2024 | 03-18-2024 | Complete | 03-18-2024 |
| #Sprint6 Manage Students | 14d | 03-19-2024 | 04-01-2024 | Complete | 04-01-2024 |
| Add student | 5d | 03-19-2024 | 03-23-2024 | Complete | 03-23-2024 |
| Delete Student | 5d | 03-24-2024 | 03-28-2024 | Complete | 03-28-2024 |
| Edit Student | 4d | 03-29-2024 | 04-01-2024 | Complete | 04-01-2024 |
| #Sprint7 Manage Exams | 4d | 04-02-2024 | 04-05-2024 | Complete | 04-05-2024 |
| Exam Schedule | 4d | 04-02-2024 | 04-05-2024 | Complete | 04-05-2024 |

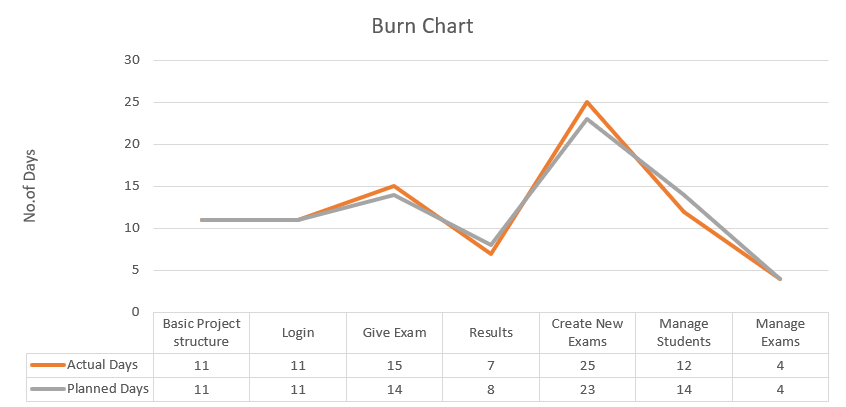
## Agile Sprint Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task Name | Story | Sprint Ready | Priority | Status | Story Point |
| #Sprint1 Basic Project structure | Yes | Yes | High | Complete | 10 |
| UI Structure | Yes | Yes | High | Complete | 9 |
| Data Base Management | Yes | Yes | High | Complete | 9 |
| #Sprint2 Login | Yes | Yes | High | Complete | 10 |
| Email and Password verification | Yes | Yes | High | Complete | 8 |
| Authentication | Yes | Yes | Medium | Complete | 7 |
| Give feedback | Yes | Yes | Medium | Complete | 9 |
| #Sprint3 Give Exam | Yes | Yes | High | Complete | 10 |
| Take Assign Exam | Yes | Yes | High | Complete | 10 |
| Answer exam question | Yes | Yes | High | Complete | 9 |
| Submit Exam Response | Yes | Yes | High | Complete | 9 |
| Can make complain if they have problem | Yes | Yes | Medium | Complete | 7 |
| #Sprint4 Results | Yes | Yes | High | Complete | 10 |
| View Results | Yes | Yes | Medium | Complete | 8 |
| Download Results | Yes | Yes | High | Complete | 9 |
| #Sprint 5 Create New Exams | Yes | Yes | High | Complete | 10 |
| Define exam parameters | Yes | Yes | High | Complete | 8 |
| Set Exam Question and Answer | Yes | Yes | High | Complete | 9 |
| Edit Existing Exams | Yes | Yes | High | Complete | 8 |
| Manage Question | Yes | Yes | High | Complete | 7 |
| #Sprint6 Manage Students | Yes | Yes | High | Complete | 10 |
| Add student | Yes | Yes | Medium | Complete | 8 |
| Delete Student | Yes | Yes | High | Complete | 8 |
| Edit Student | Yes | Yes | Medium | Complete | 9 |
| #Sprint7 Manage Exams | Yes | Yes | High | Complete | 10 |
| Exam Schedule | Yes | Yes | High | Complete | 10 |

## Agile Test Plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project Name | | Examination System | | | Device | | HP intel core i5  12th gen | |
| Test Case Id | | 1 | | | Test Title | | **Testing whole system** | |
| Module Name | | Manage Exam System | | | Tested By | | **Fahzain, Sarjil** | |
| Priority | | High | | | Execution Date | | 03-04-2024 | |
| Test | Test Step | | Action | Expected Result | | Actual Result | | Pass |
| 1 | **Click on “Start exam button”** | | **The exam must be going to start** | **Exam started successfully** | | **Exam started successfully** | | Yes |
| 2 | **Click on “Cancel button”** | | **The exam must be going to cancel** | **Exam Cancelled** | | **Exam Cancelled** | | **Yes** |
| 3 | **Click on ”Paper upload button”** | | **Exam Paper Must Upload** | **Exam Paper uploaded successfully** | | **Exam Paper uploaded successfully** | | **Yes** |
| 4 | **Click on ”Click on exam button”** | | **Student exam wants to be started** | **Exam started** | | **Exam started** | | Yes |
| 5 | **Click on “Result download button”** | | **Result must be download** | **Download successfully** | | **Download successfully** | | Yes |
| 6 | **Click on “Block account button”** | | **Student account must be blocked** | **Blocked successfully** | | **Blocked successfully** | | Yes |

## Earned-value and burn charts



# Proposed Enhancements

Feature Expansion: Identify and prioritize additional features based on user feedback and requirements. Implement new features to streamline exam scheduling, grading, and reporting processes.

Performance Optimization: Analyze system performance to identify and address bottlenecks. Optimize code and database queries to improve system responsiveness and speed.

User Interface Refinement: Gather user feedback to identify areas for UI improvement. Refine UI design for intuitive navigation and enhanced user experience.

Mobile Responsiveness: Ensure the system is fully responsive and functions seamlessly on mobile devices.

Test compatibility across various screen sizes and browsers.

Scalability Planning: Assess system scalability requirements for handling increased user load. Implement scalability solutions to accommodate growth without compromising performance.

# Conclusion

* These changes are meant to make our Exam Management System work better, be easier to use, and keep everything safe. This will make it easier for teachers, students, and administrators to use without any problems.

# Bibliography

1. MongoDB Documentation: MongoDB's official documentation offers extensive resources for developers working with MongoDB, a NoSQL database used in the MERN stack. It includes guides, tutorials, and references for MongoDB's features and functionalities. Available at: MongoDB Documentation.
2. Express.js Documentation: Express.js is a web application framework for Node.js, often used in combination with MongoDB in the MERN stack. The Express.js documentation provides guidance on building web applications and APIs using Express.js. Available at: Express.js Documentation.
3. React Documentation: React's official documentation is an essential resource for developers building user interfaces with React.js, a front-end JavaScript library in the MERN stack. It offers guides, tutorials, and API references for React development. Available at: React Documentation.
4. Node.js Documentation: Node.js is a server-side JavaScript runtime environment used in the MERN stack. The Node.js documentation provides information on using Node.js to build scalable and efficient web applications. Available at: Node.js Documentation.