Cairo University
Faculty of Computers and Artificial Intelligence
Department of Computer Sciences

Our School

Supervised By

Dr. Basher Abdelfattah

TA. Menna Youssef

Implemented By

20200155	Hassan Ali Mohamed Ali Mohamed
20201010	Ahmed Abdelkarim Ahmed
20201060	Donia Ahmed Abozeid
20201208	Heba Abdelwahab Said

Graduation Project

Academic Year 2024-2025

Final Documentation

Table of Contents

List of Figures **4**

List of Tables **5**

List of Abbreviations **5**

Chapter 1: Introduction **6**

1.1. Motivation *6*

1.2. Problem Definition *6*

1.3. Project Objective (Suggested Solution) *7*

1.4. Gantt Chart of Project Time Plan *8*

1.5. Project Development Methodology *9*

1.6. The used tools in the project (SW and HW) *9*

1.7. Report Organization (Summary of the Rest of the Report) *10*

Chapter 2: Related Work **11**

2.1. Overview of Existing Educational Websites *11*

2.2. Analysis of Features *12*

2.3. Technologies Used *13*

2.4. Summary and Conclusion *14*

Chapter 3: System Analysis **15**

3.1. Project Specifications *15*

3.1.1. Functional Requirement *15*

3.1.2. Non-functional Requirement *16*

3.2. Use Case Diagrams *17*

3.3. Use Case Table *18*

Chapter 4: System Design **23**

4.1. System Component Diagram *23*

4.2. System Class Diagrams *23*

4.3. Sequence Diagrams *24*

4.4. Project ERD *28*

4.5. System GUI Design *29*
4.5.1. Admin Pages *29*
Chapter 5: Implementation and Testing **34**
5.1. Implementation *34*
5.2. Testing *35*
5.2.1. Back-End *35*
5.2.2. Front-End *51*
References **54**

List of Figures

Figure 1: Use Case Diagram 17
Figure 2: Component Diagram 23
Figure 3: Login 24
Figure 4: Get Student Announcements 25
Figure 5: Teacher Make Announcement to Class 25
Figure 6: Teacher Take Attendance 26
Figure 7: Teacher Upload Material 26
Figure 8: Student View Material 27
Figure 9: Parent View Student Progress Report 27
Figure 10: ERD Component 28
Figure 11: Add New Student 29
Figure 12: Students 29
Figure 13: Attendance 30
Figure 14: Grades 30
Figure 15: Parents 31
Figure 16: Classes 31
Figure 17: Subjects 32
Figure 18: Announcements 32
Figure 19: Add Announcement 33
Figure 20: Add Evaluation Report 33
Figure 21: User Login with Valid Data 36
Figure 22: User login with wrong email 36
Figure 23: User login with wrong password 37
Figure 24: Add Parent with valid data 39
Figure 25: Add Parent with first name only 39
Figure 26: Add Parent with invalid Gmail 40
Figure 27: Add Parent with phone number less than 11 numbers 40
Figure 28: Add Parent with phone number less than 11 numbers 41
Figure 29: Admin Sends a valid Announcement 43
Figure 30: Admin sends an Announcement with empty title 43
Figure 31: Admin sends an Announcement with empty Body 44
Figure 32: Admin sends an Announcement with invalid number 44
Figure 33: Admin chooses dept and enter no. of Class that not Exist in level 46
Figure 34: Admin chooses dept and enter no. of Class that Already Exist in level 46
Figure 35: Admin Enters Valid Degrees for Subjects 48

Figure 36: Admin Enters Invalid Degrees for Subjects 48
Figure 37: Admin Enters Subject Record that wasn't repeated 50
Figure 38: Admin Enters Subject Record that was repeated 50
Figure 39: Wrong Password 51
Figure 40: Wrong Email 51
Figure 41: Parent Validation 52
Figure 42: Parent Successfully Added 52
Figure 43: Announcement Validation 53
Figure 44: Announcement Successfully Post 53

List of Tables

Table 1: Use Case 1 — Login 18
Table 2: Use Case 2 — Take Attendance 19
Table 3: Use Case 3 — Assign Grades 20
Table 4: Use Case 4 — Add Student 21
Table 5: Use Case 5 — Send Announcement 22
Table 6: Test Table 1 — Login User 35
Table 7: Test Table 2 — Add Parent 38
Table 8: Test Table 3 — Send Announcement 42
Table 9: Test Table 4 — Add Class 45
Table 10: Test Table 5 — Update Final Degree of Subjects for Student 47
Table 11: Test Table 6 — Add Subject Record 49

List of Abbreviations

ng	Angular	Dept	Department
SW	Software	SPA	Single-Page Application
HW	Hardware		
ML	Machine Learning		
No	Number		

Chapter 1: Introduction

1.1. Motivation

In the realm of education, effective communication among schools, teachers, students, and parents is pivotal for creating a conducive learning environment. Traditional methods of communication have become outdated, leading to a communication gap. To address this, our proposed web platform aims to serve as a centralized hub, allowing schools to make announcements, teachers to interact with students and parents, students to access attendance records and academic materials, and parents to monitor their children's progress. Our solution seeks to streamline communication and information sharing within the school community through a user-friendly and comprehensive platform. The Technologies that will be used in our project are frontend (Angular), Back-end (ASP.NET), Database (Microsoft SQL Server).

1.2. Problem Definition

Schools often lack a centralized platform for making announcements, resulting in fragmented communication channels. This can lead to a lack of timely and consistent information reaching students, parents, and teachers, impacting their ability to stay informed about crucial updates, events, and changes within the school community.

The limitations in teacher-student interaction beyond the classroom create challenges in providing additional support and clarification. Without a dedicated platform, teachers may find it challenging to communicate effectively with students outside of formal class hours, hindering the overall learning experience.

Additionally, the existing communication channels between teachers and parents are often inadequate. A lack of direct and transparent communication can lead to misunderstandings about a student's progress,

making it essential to establish more direct and accessible lines of communication between these stakeholders.

The inefficiency in distributing study materials poses another challenge. Teachers may face difficulties in ensuring that students have timely access to necessary resources, impacting the learning continuity. This is particularly crucial in the current educational landscape, where digital access to materials is becoming increasingly important.

Furthermore, the lack of immediate access to student grades and progress reports for parents can hinder their ability to actively engage in their child's academic journey. Providing a transparent and real-time view of a student's academic performance is crucial for parental involvement and support.

1.3. Project Objective (Suggested Solution)

To address these challenges, we decided to make a centralized communication platform for schools, providing the following solutions.

Centralized Announcement System:

The website will feature a centralized announcement system, ensuring that crucial updates, events, and changes are disseminated in a timely and consistent manner to students, parents, and teachers. This eliminates the fragmentation in communication channels.

Teacher-Student Interaction Platform:

A dedicated space within the platform will facilitate teacher-student interaction beyond the confines of the traditional classroom. This feature allows teachers to provide additional support, clarification, and engage with students effectively outside of formal class hours, thereby enhancing the overall learning experience.

Direct and Transparent Teacher-Parent Communication:

The platform will establish direct and transparent communication channels between teachers and parents. This ensures that parents have immediate access to their child's progress, eliminating misunderstandings and fostering a collaborative environment for the student's academic journey.

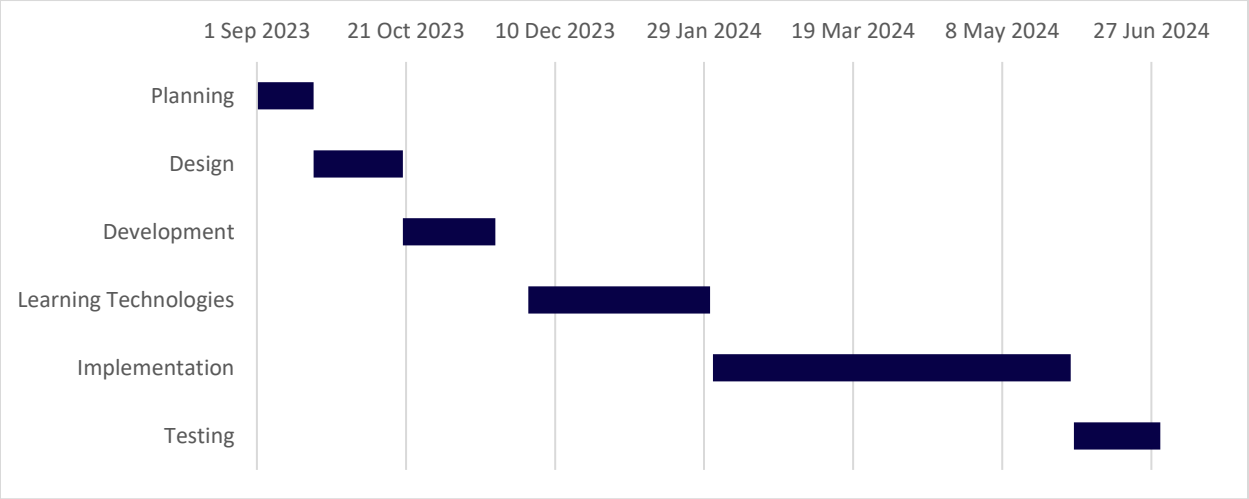
Efficient Distribution of Study Materials:

Teachers will be equipped with tools to efficiently upload and distribute study materials digitally. This addresses the challenge of ensuring timely access to necessary resources, promoting learning continuity in an era where digital access to materials is increasingly crucial.

Real-time access to Grades, attendance records, and Progress Reports:

Parents will have a dedicated portal providing real-time access to student grades attendance recodes and progress reports. This feature empowers parents to actively engage in their child's academic journey, offering a transparent view of academic performance and facilitating timely interventions if needed.

1.4. Gantt Chart of Project Time Plan



1.5. Project Development Methodology

For this project, we are utilizing the Agile development methodology. This approach allows for iterative development, where the project is broken down into smaller cycles called sprints. Each sprint involves planning, executing, and evaluating the progress, which allows for continuous improvement and adaptation based on feedback.

Key elements of Agile methodology:

- **Iterations:** Development cycles that typically last 2-4 weeks.
- **Scrum Meetings:** Daily stand-up meetings to discuss progress and obstacles.
- **Backlog:** A list of tasks and features to be completed.
- **User Stories:** Descriptions of features from the end-user perspective.
- **Retrospective:** Meetings held at the end of each sprint to evaluate what went well and what can be improved.

1.6. The used tools in the project (SW and HW)

Software (SW):

- **Angular:** Framework for building the frontend of the application.
- **ASP.NET Core:** Framework for developing the backend.
- **Microsoft SQL Server:** Database management system for storing data.
- **Visual Studio Code:** IDE for writing and editing code.
- **Git:** Version control system for managing code changes.
- **Postman:** API testing tool.
- **Jira:** Project management and issue tracking tool.
- **Node.js:** For running Angular applications during development.

Hardware (HW):

- **Development Machines:** Computers with at least 8GB RAM, Intel i5 processor or higher.
- **Servers:** Web servers for hosting the application, preferably cloud-based solutions like AWS or Azure.
- **Network Infrastructure:** Reliable internet connection for team collaboration and access to cloud services.

1.7. Report Organization (Summary of the Rest of the Report)

The rest of the document discusses the different phases that describe and illustrate the characteristics of the project.

Chapter Two: Related Works, here we compare other websites that offer same services to our project.

Chapter Three: System Analysis, including functional requirements of the system according to what we found in related works, non-functional requirements that indicate the performance of the system, use case diagram that shows stakeholders and also use case scenarios

Chapter Four: System Design, include all diagrams that describe relations between classes and component in system:

- Component Diagram.
- Class Diagram.
- Sequence Diagrams.
- Entity Relation Diagram.
- GUI design.

Chapter Five: Implementation and Testing include Back-end and testing scenarios results and front end with their test scenarios also

Chapter 2: Related Work

In this chapter, we review existing educational websites and systems, focusing on their features, technologies used, and the benefits they offer to different stakeholders such as students, parents, teachers, and administrators. This analysis helps identify the strengths and weaknesses of current solutions and provides a foundation for developing a more effective and user-friendly system.

2.1. Overview of Existing Educational Websites

1. **Google Classroom**
    - **Features:** Assignment distribution, real-time collaboration, grading, and feedback.
    - **Technologies Used:** Google Cloud Platform, React, Firebase.
    - **Strengths:** Integration with Google services, ease of use, real-time collaboration.
    - **Weaknesses:** Limited customization options, dependency on Google ecosystem.

2. **Edmodo**
    - **Features:** Social learning platform, communication tools, assignment management, quizzes.
    - **Technologies Used:** Ruby on Rails, PostgreSQL, Redis.
    - **Strengths:** Social media-like interface, strong community support, wide range of resources.
    - **Weaknesses:** Privacy concerns, potential for distractions due to social features.

3. **Moodle**
    - **Features:** Course management, customizable learning environments, assessments, forums.
    - **Technologies Used:** PHP, MySQL, Apache.
    - **Strengths:** Highly customizable, open-source, extensive plugin ecosystem.
    - **Weaknesses:** Steeper learning curve, requires significant setup and maintenance.

4. **Blackboard**
    - **Features:** Course management, virtual learning environments, assessments, analytics.
    - **Technologies Used:** Java, Oracle Database.
    - **Strengths:** Comprehensive features, robust analytics, widely adopted in higher education.
    - **Weaknesses:** Expensive, complex interface, requires extensive training.

2.2. Analysis of Features

- **User Management:**

  Most platforms offer comprehensive user management systems for students, teachers, and administrators. Role-based access control is a common feature to ensure appropriate permissions and access levels.

- **Communication and Collaboration:**

  Real-time communication tools such as chat, forums, and video conferencing are crucial for effective collaboration. Integration with third-party tools like Google Meet or Zoom is often implemented.

- **Content Management:**

  Efficient content management systems to upload, organize, and distribute educational materials. Support for various file types including documents, videos, and interactive content.

- **Assessment and Feedback:**

  Tools for creating and managing assignments, quizzes, and exams. Automated grading systems and detailed feedback mechanisms to support student learning.

- **Analytics and Reporting:**

  Data analytics tools to track student performance and engagement. Customizable reports for teachers and administrators to monitor progress and identify areas for improvement.

2.3. Technologies Used

- **Frontend Technologies:**

  In the frontend, we chose Angular for its powerful features and support for building robust single-page applications (SPAs). Angular offers a structured framework, excellent TypeScript integration, and extensive libraries that

facilitate efficient development and maintenance of complex user interfaces. Its component-based architecture also promotes code reusability and scalability, making it a suitable choice for modern frontend development.

- **Backend Technologies:**

  We used ASP.NET Core and SQL Server for backend development for their robustness and compatibility. ASP.NET Core provides a modern, cross-platform framework with strong community support and performance optimizations. SQL Server offers reliability, scalability, and comprehensive data management capabilities, making it ideal for handling structured data efficiently in backend applications

2.4. Summary and Conclusion

The analysis of existing educational websites reveals a variety of features and technologies that cater to the diverse needs of educational institutions. While many platforms offer comprehensive solutions, there are opportunities for improvement, particularly in terms of customization, ease of use, and integration with other educational tools.

Our project aims to leverage the strengths of these existing solutions while addressing their limitations. By utilizing modern technologies such as Angular for the frontend and ASP.NET for the backend, we aim to create a user-friendly, scalable, and efficient educational website that meets the needs of students, parents, teachers, and administrators.

Chapter 3: System Analysis

3.1. Project Specifications

3.1.1. Functional Requirement

For Admins:

- login.
- Add students, parents, teachers.
- Assign students to class.
- Edit information of students, parents, and teachers.
- View all students, parents, and teachers and enable search for them.
- Assign final degrees for all students.
- View attendance of all students and send Absence warnings.
- Add subjects.
- Add classes and assign teachers in a class with specific subject.
- Send request meeting to parents.
- Send announcements to all users (parents, teacher, student).

For Teachers:

- Login.
- Upload materials.
- View all the classes they teach.
- Assign degrees for each class.
- Take attendance in a class.
- Send a progress report about student to parent.
- Send request meeting to parents.
- View all parents of their students.
- Send announcements to classes.
- Chat with parents and students.

For Students:

- Login.
- View all their subjects and materials.
- View all teachers that teach them.
- View their attendance record and absent warnings.
- View school and their class announcements.
- View their grades.
- Chat with their teachers.

For Parents:

- Login.
- View progress reports of their children's subjects.
- View information about their children's teachers.
- View grade for each child.
- View announcement.
- Chat with the teachers of their child.

3.1.2. Non-functional Requirement

- **Usability:** The application should be user-friendly, with a clear and intuitive interface.

- **Maintainability**: The application should be easy to maintain and update. It should follow coding standards and have proper documentation.

- **Security:** The application should use strong authentication methods to verify user identities. Sensitive data, such as student information and private messages, should be encrypted.

- **Reliability:** The application should ensure the accuracy and consistency of data, with mechanisms to prevent data loss or corruption. The application should gracefully handle errors and provide clear and helpful messages to users.

- **Availability:** The application should be available 99.5% of the time during school hours and business days.

- **Response time:** The application should load pages and respond to user actions within 2 seconds on average.

3.2. Use Case Diagrams

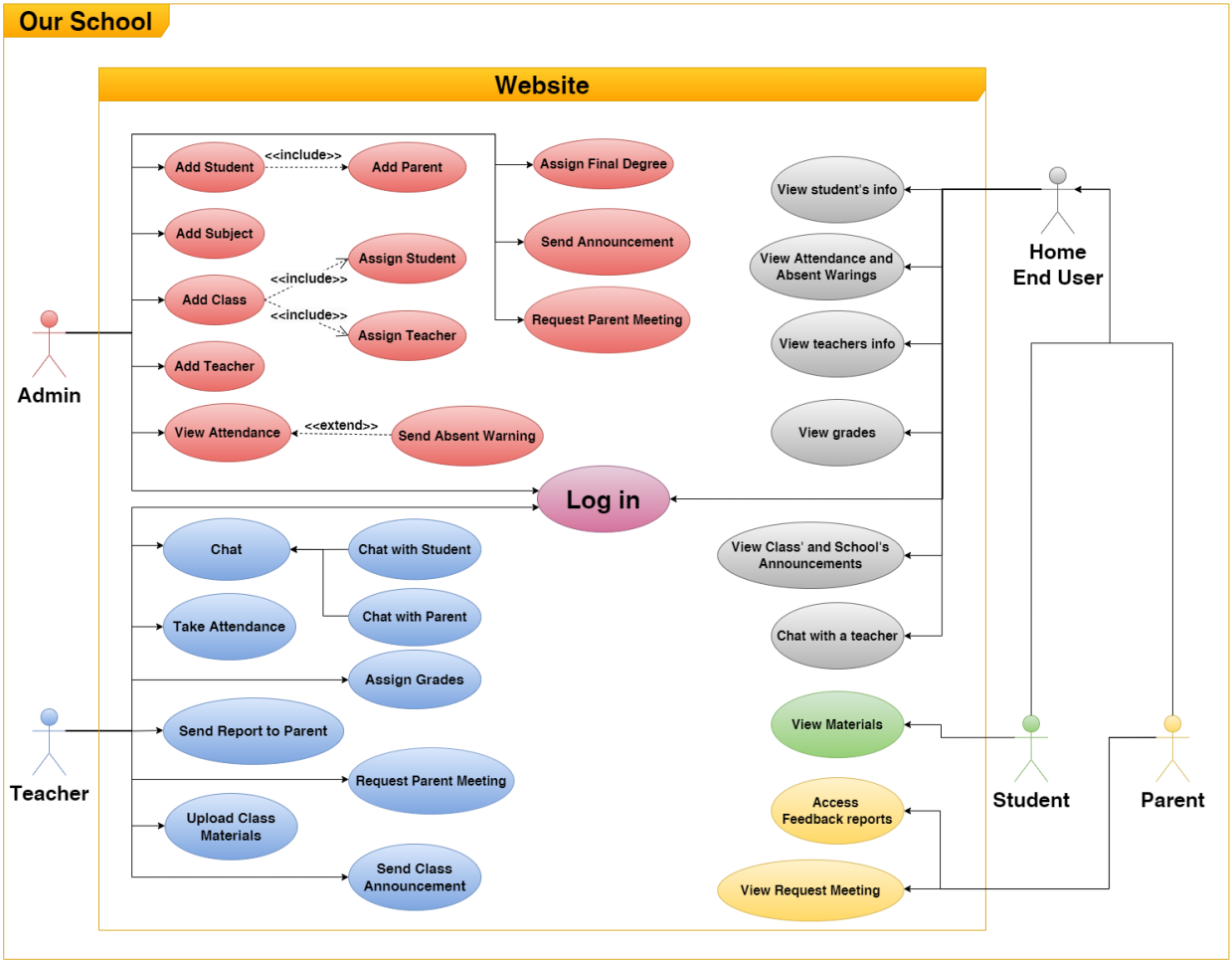


Figure 1: Use Case Diagram

3.3. Use Case Table

Use Case ID:	#1
Use Case Name:	Login
Actors:	User (Student, Parent, Teacher)
Description:	The user logs into the platform using their credentials.
Pre-conditions:	The user must have an existing account with valid login credentials. The login page is accessible.
Post-conditions:	The user is successfully authenticated and redirected to their respective dashboard.
Flow of Events:	1. The user navigates to the login page. 2. The user enters their username and password. 3. The user submits the login form. 4. The system validates the input against stored credentials. 5. If the credentials are correct, the user is authenticated. 6. The user is redirected to their respective dashboard (Student, Parent, or Teacher). 7. If the credentials are incorrect, an error message is displayed, and the user is prompted to try again.

Table 1: Use Case 1 — Login

Use Case ID:	#2
Use Case Name:	Take Attendance
Actors:	User (Teacher)
Description:	The teacher records attendance for a class session.
Pre-conditions:	The teacher is logged into the system. The class and student information are available.
Post-conditions:	The attendance record is updated in the system.
Flow of Events:	1. The teacher navigates to the attendance section. 2. The teacher selects the class for which attendance is to be taken. 3. The system displays a list of students in the selected class. 4. The teacher marks each student as present or absent. 5. The teacher submits the attendance record. 6. The system saves the attendance record. 7. The system confirms that attendance has been successfully recorded.

Table 2: Use Case 2 — Take Attendance

Use Case ID:	#3
Use Case Name:	Assign Grades
Actors:	User (Teacher)
Description:	The teacher assigns grades to students for a particular assignment or exam.
Pre-conditions:	The teacher is logged into the system. The assignment or exam information is available.
Post-conditions:	The grades are saved and updated in the system. Students can view their grades.
Flow of Events:	1. The teacher navigates to the grading section. 2. The teacher selects the class and the specific assignment or exam to grade. 3. The system displays a list of students and fields to enter grades. 4. The teacher enters grades for each student. 5. The teacher submits the grades. 6. The system saves the grades. 7. The system confirms that the grades have been successfully recorded. 8. The students are notified that their grades are available.

Table 3: Use Case 3 — Assign Grades

Use Case ID:	#4
Use Case Name:	Add Student
Actors:	User (Admin)
Description:	The admin adds a new student to the system.
Pre-conditions:	The admin is logged into the system. The necessary student information is available.
Post-conditions:	The new student is added to the system. The student receives login credentials
Flow of Events:	1. The admin navigates to the student management section. 2. The admin selects the option to add a new student. 3. The system displays a form to enter the student's information (e.g., name, email, grade level). 4. The admin fills in the required information. 5. The admin submits the form. 6. The system validates the input. 7. The system creates a new student record. 8. The system generates and sends login credentials to the student's email. 9. The admin is notified that the student has been successfully added

Table 4: Use Case 4 — Add Student

Use Case ID:	#5
Use Case Name:	Send Announcement
Actors:	User (Admin)
Description:	The admin sends an announcement to all or specific users on the platform
Pre-conditions:	The admin is logged into the system. The announcement content is prepared.
Post-conditions:	The announcement is delivered to the intended recipients.
Flow of Events:	1. The admin navigates to the announcements section. 2. The admin selects the option to create a new announcement. 3. The system displays a form to enter the announcement details (e.g., title, message, recipients). 4. The admin fills in the required information. 5. The admin selects the recipients (e.g., all users, specific groups). 6. The admin submits the announcement. 7. The system validates the input. 8. The system sends the announcement to the selected recipients. 9. The admin receives confirmation that the announcement has been successfully sent.

Table 5: Use Case 5 — Send Announcement

Chapter 4: System Design

4.1. System Component Diagram

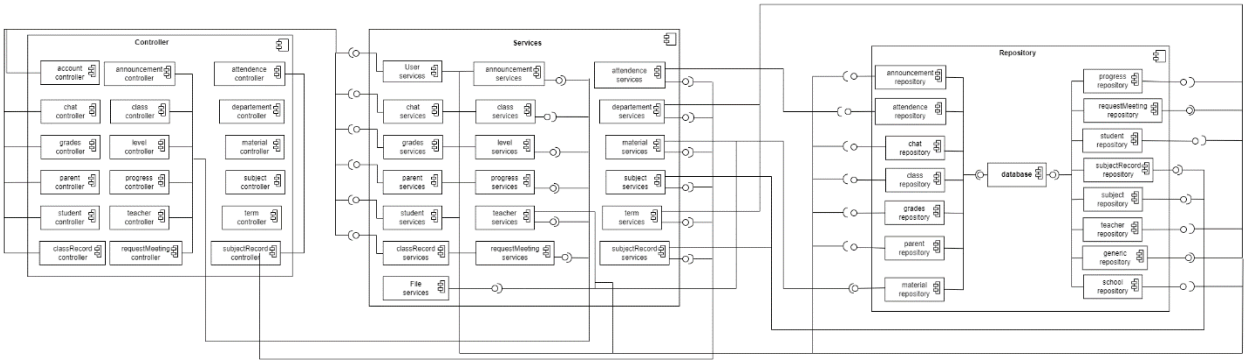


Figure 2: Component Diagram

4.2. System Class Diagrams

4.3. Sequence Diagrams

Login

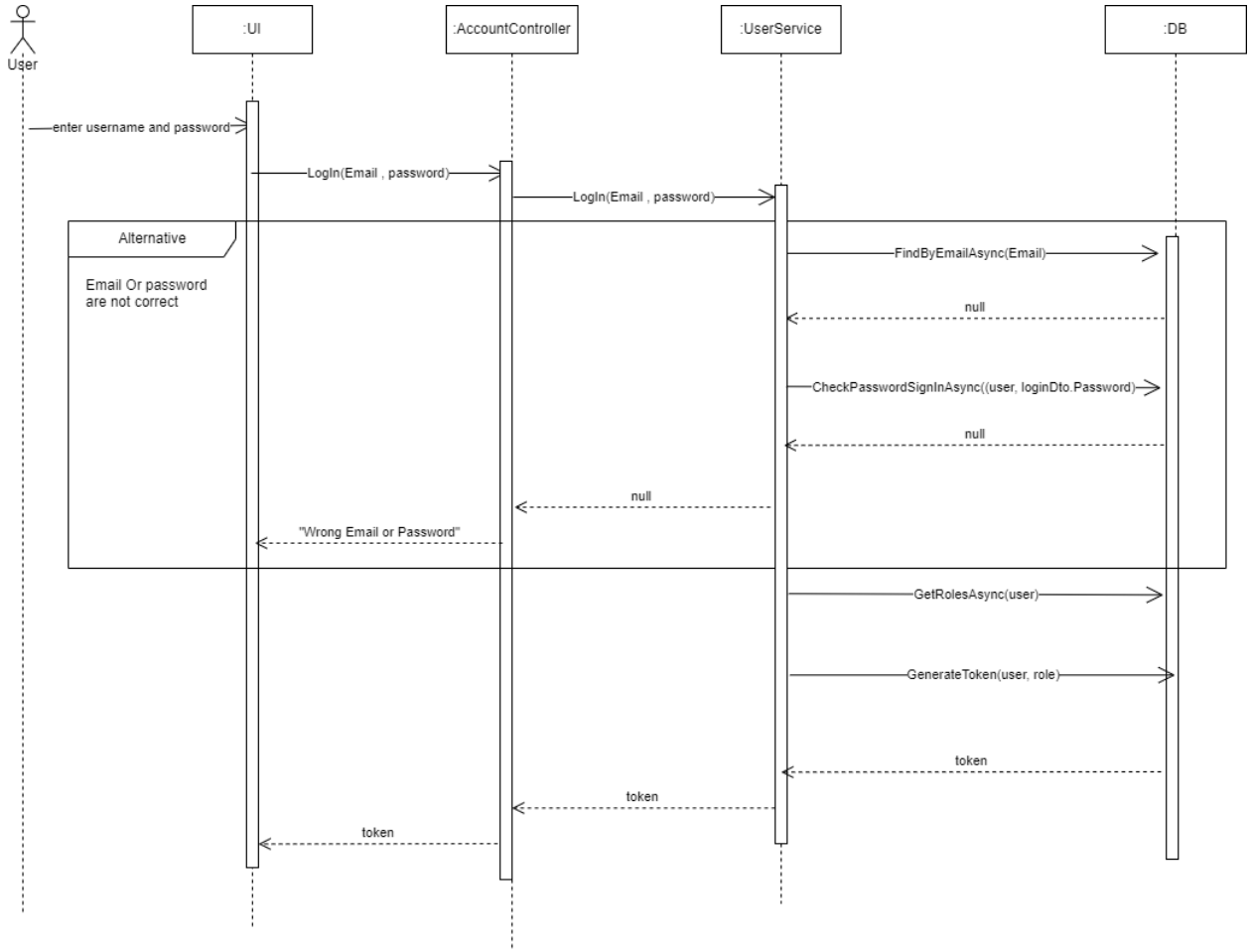


Figure 3: Login

Get Student Announcements

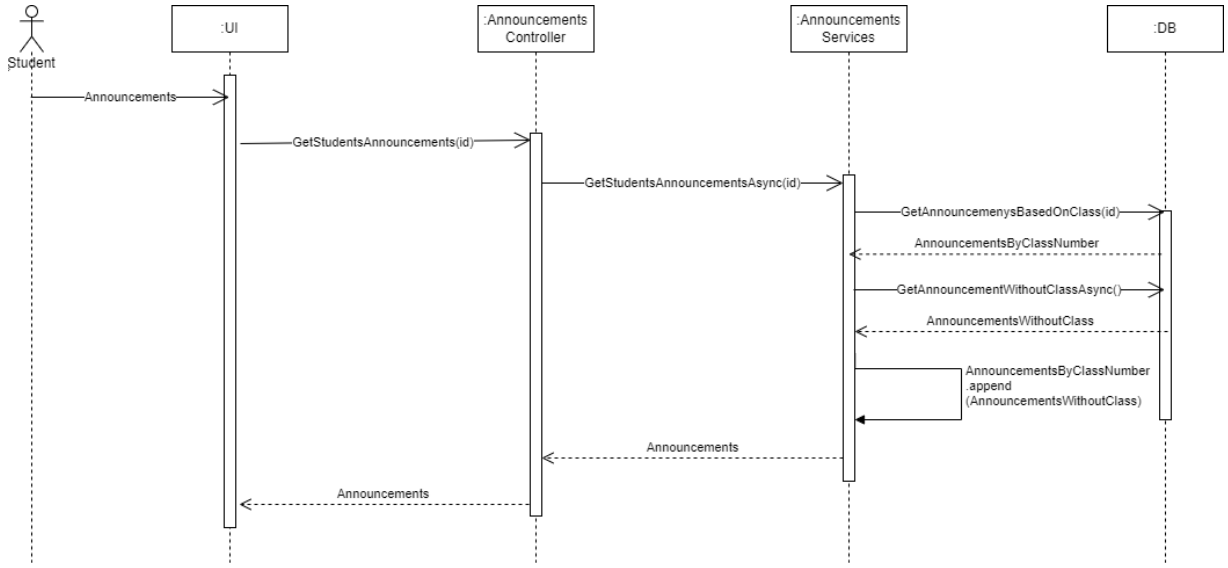


Figure 4: Get Student Announcements

Teacher Make Announcement to Class

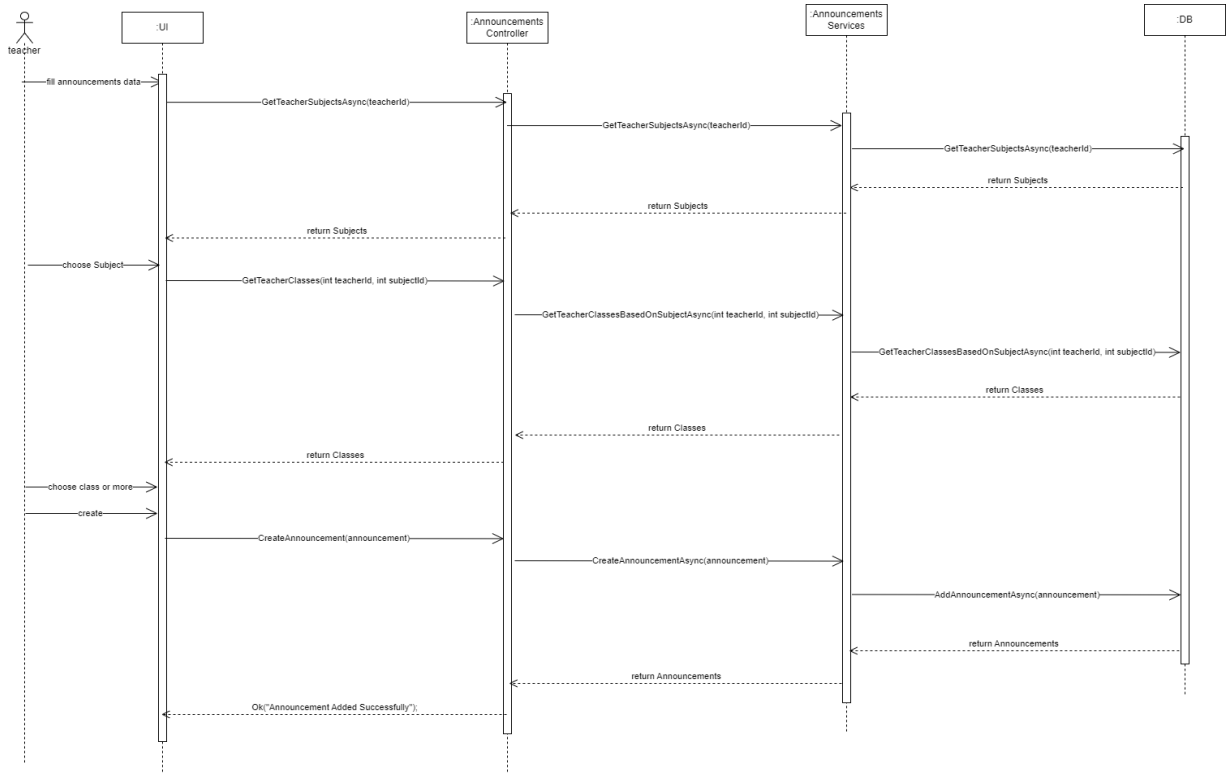


Figure 5: Teacher Make Announcement to Class

Teacher Take Attendance

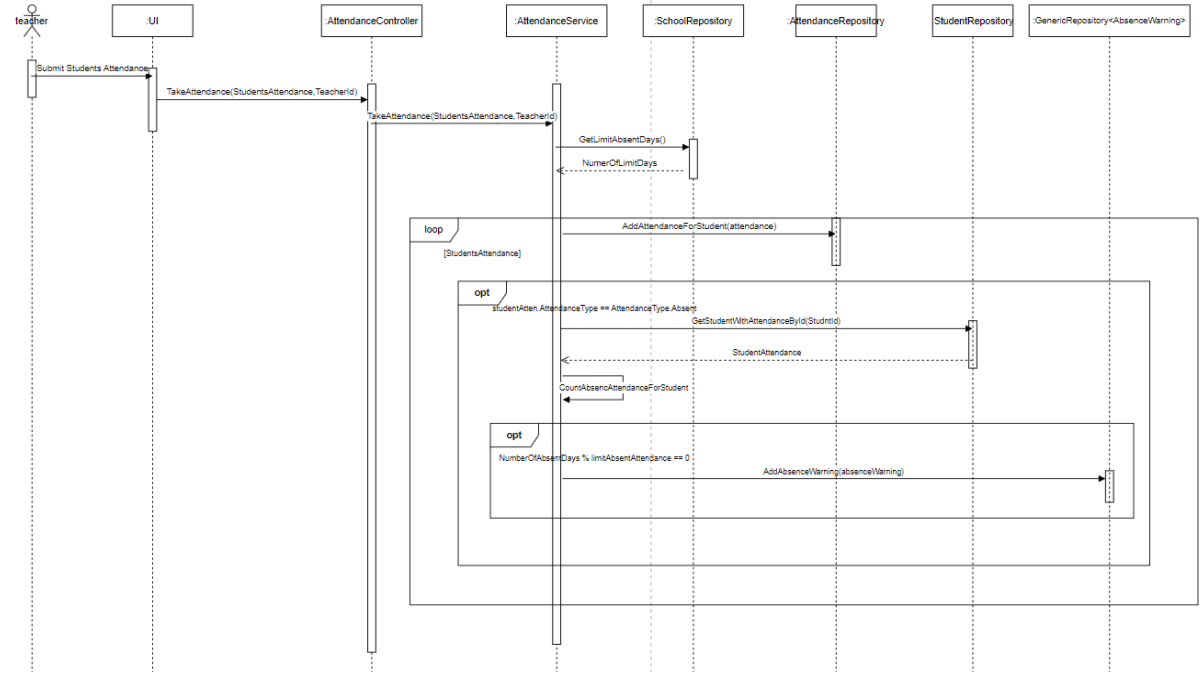


Figure 6: Teacher Take Attendance

Teacher Upload Material

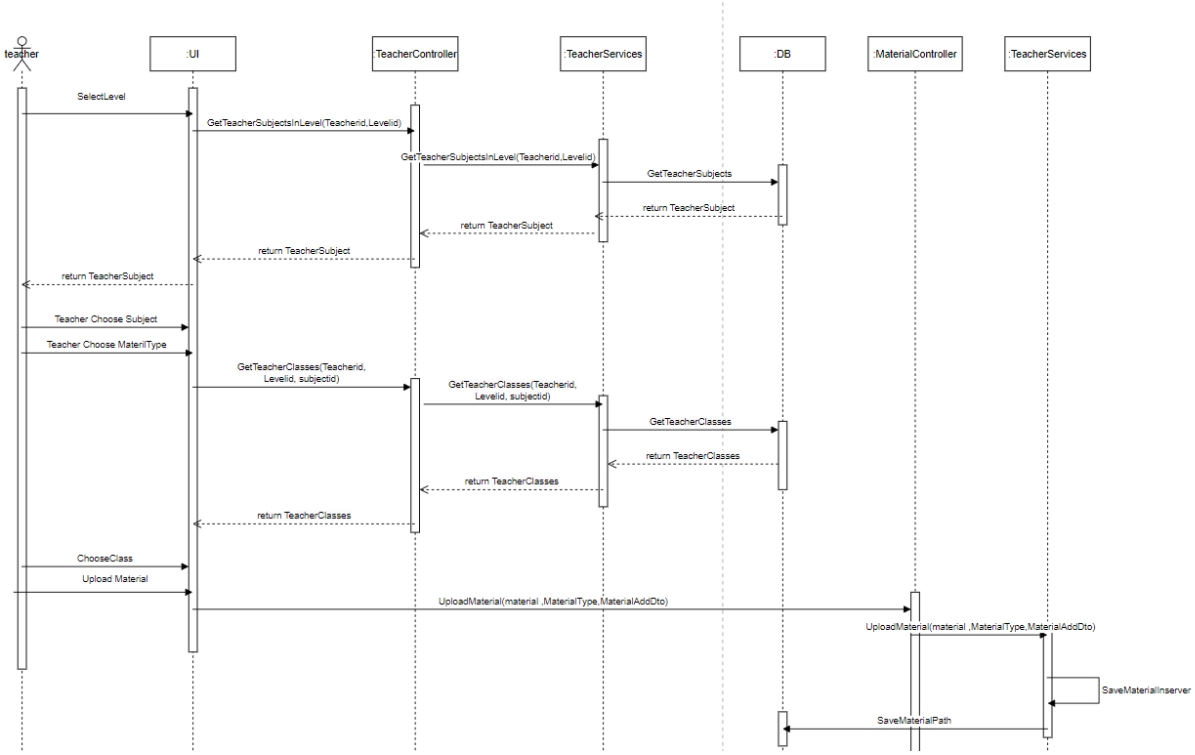


Figure 7: Teacher Upload Material

Student View Material

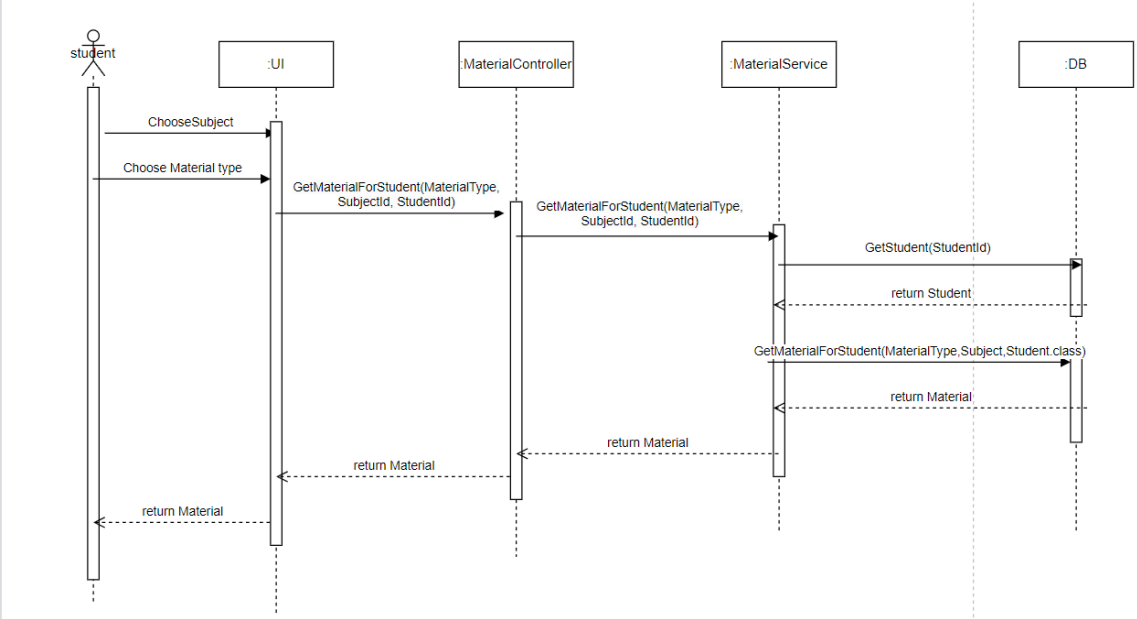


Figure 8: Student View Material

Parent View Student Progress Report

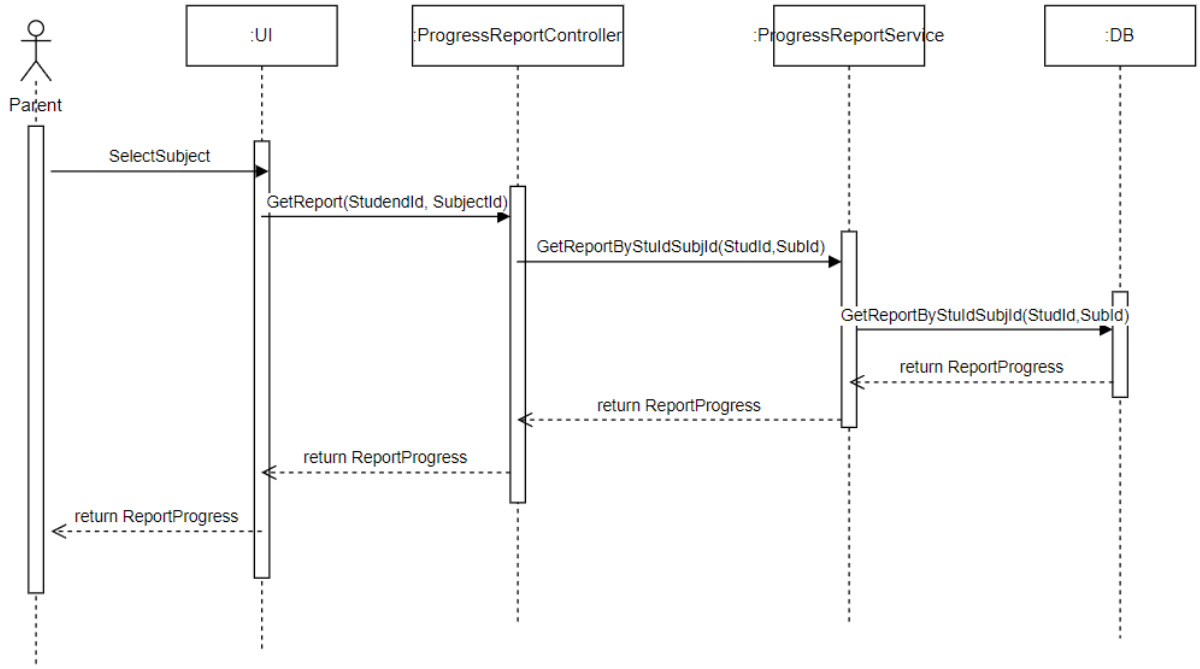


Figure 9: Parent View Student Progress Report

4.4. Project ERD

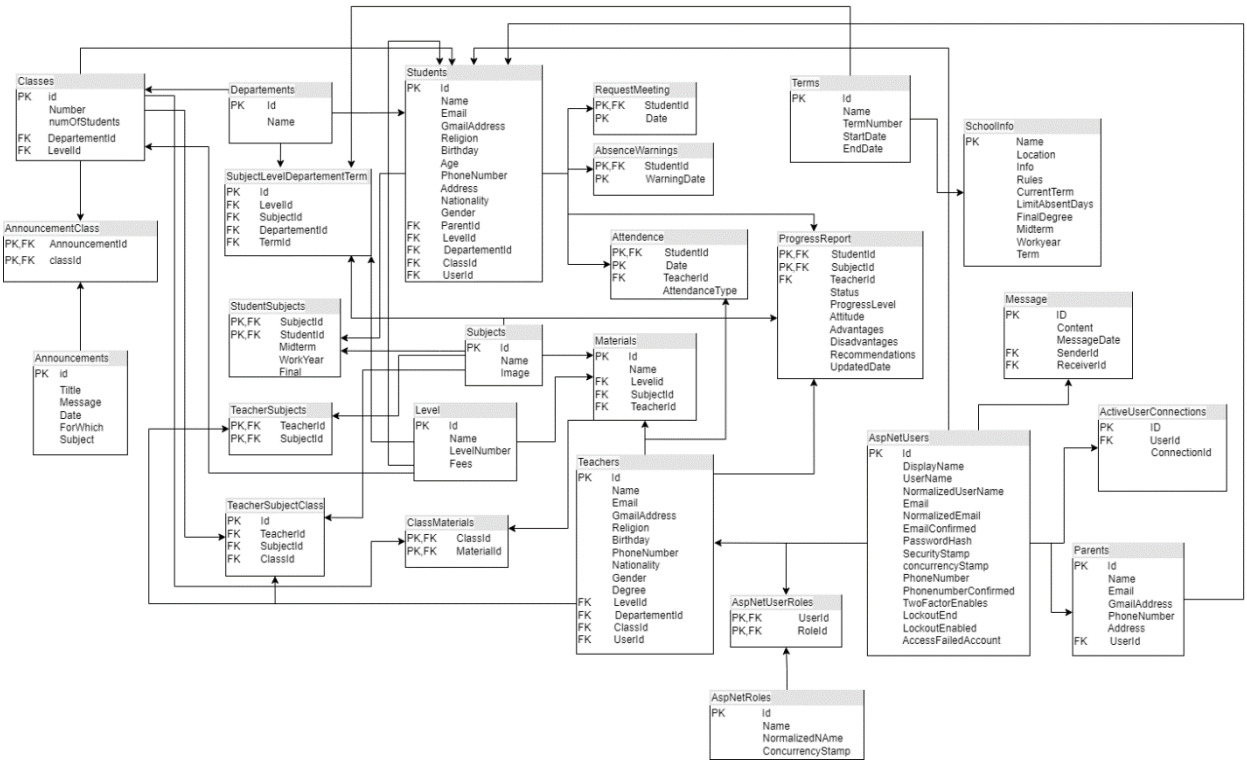


Figure 10: ERD Component

4.5. System GUI Design

4.5.1. Admin Pages

Add New Student

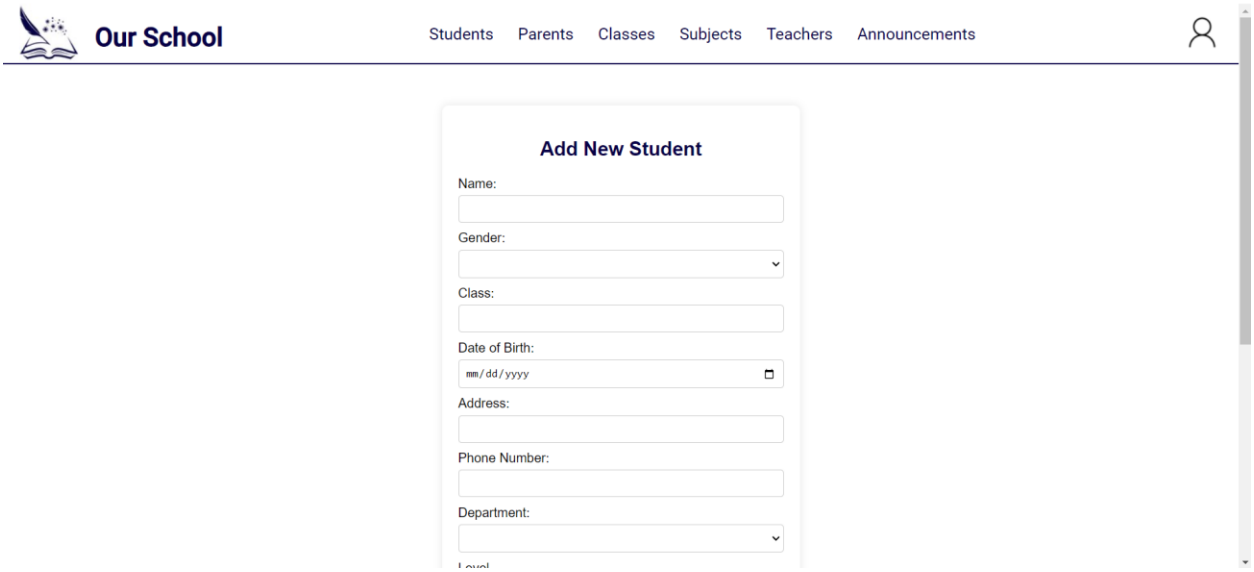


Figure 11: Add New Student

Students

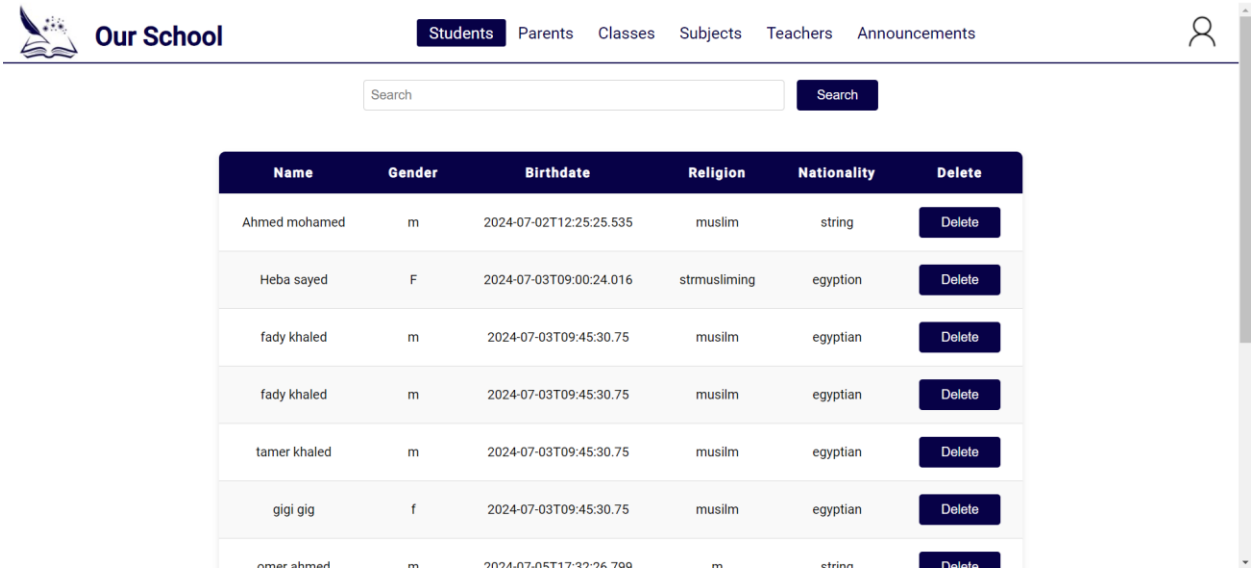


Figure 12: Students

Attendance

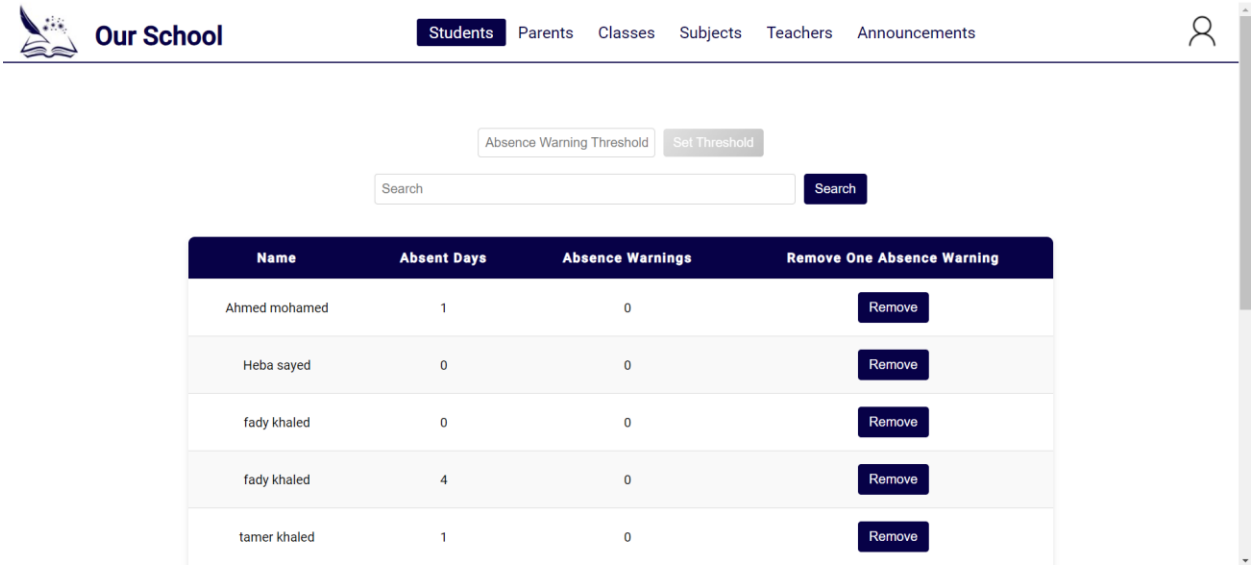


Figure 13: Attendance

Grades

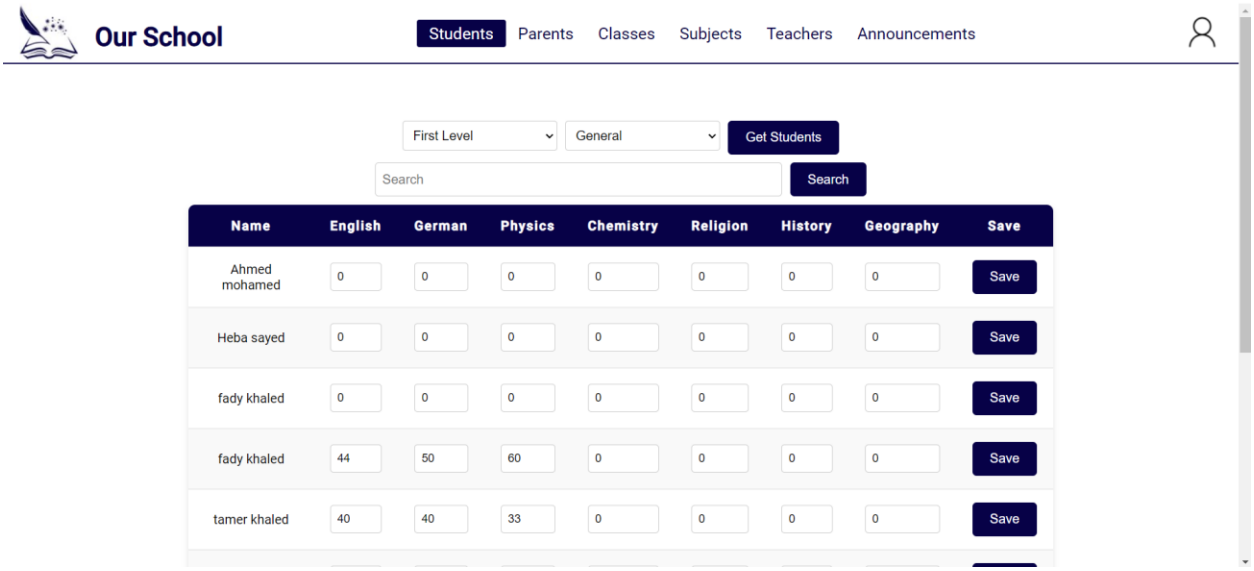


Figure 14: Grades

Parents

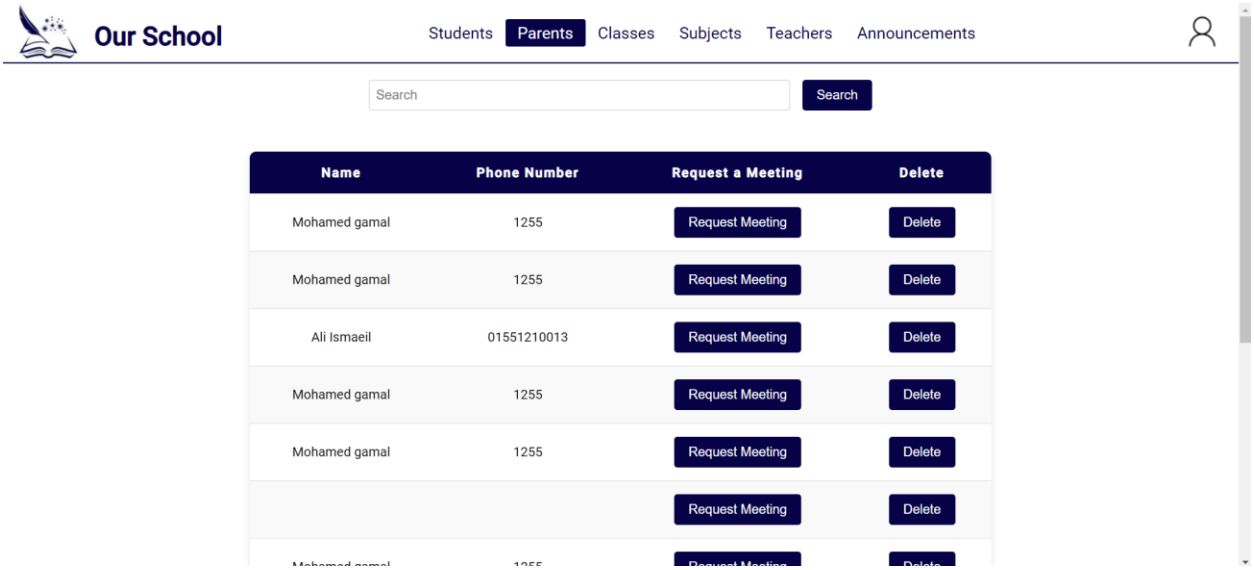


Figure 15: Parents

Classes

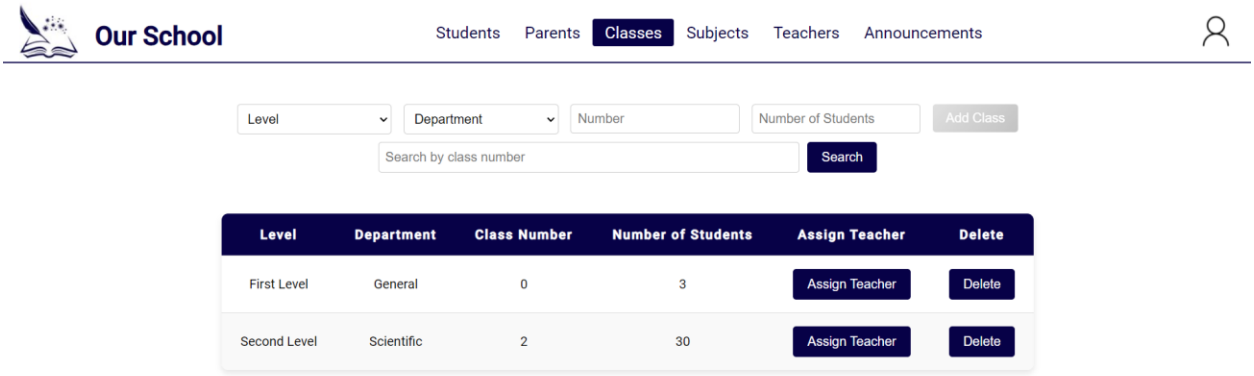


Figure 16: Classes

Subjects

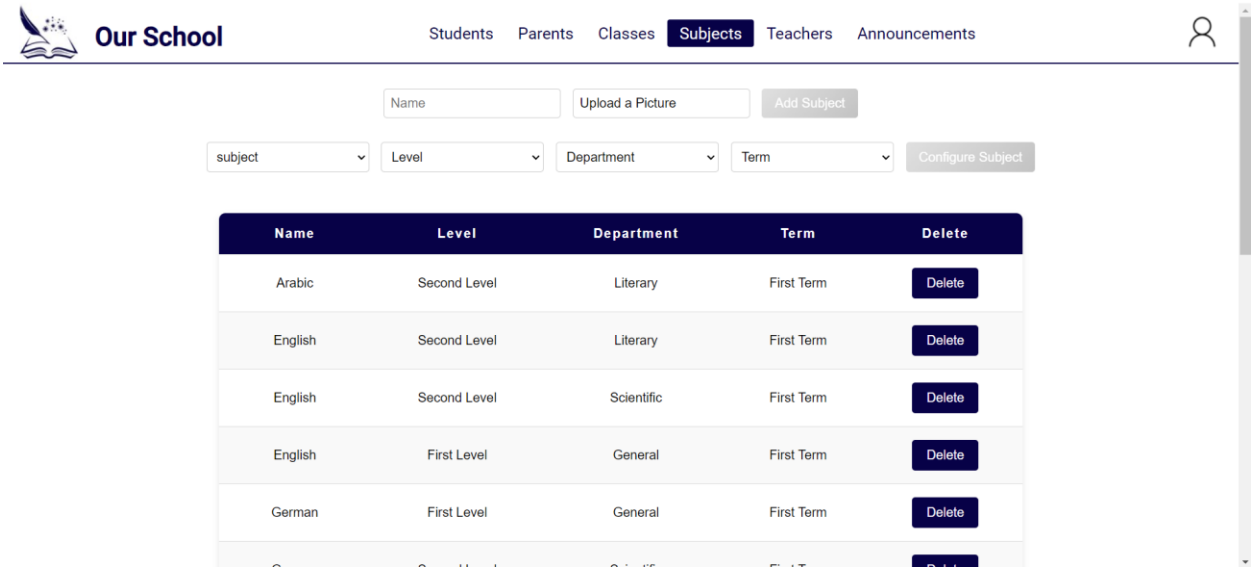


Figure 17: Subjects

Announcements

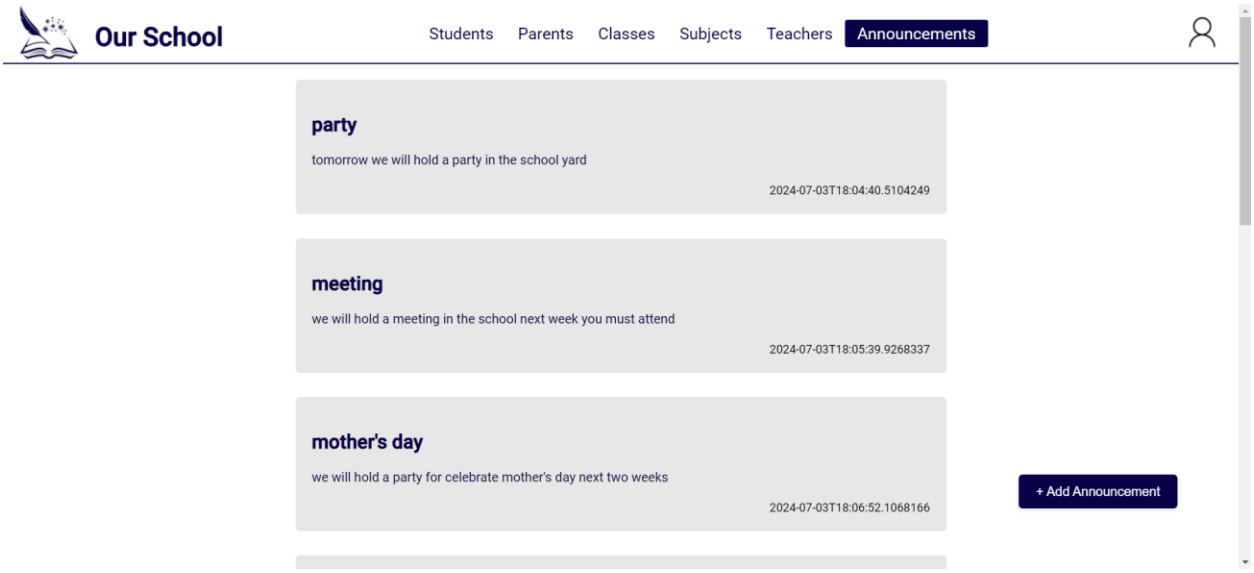


Figure 18: Announcements

Add Annoncement

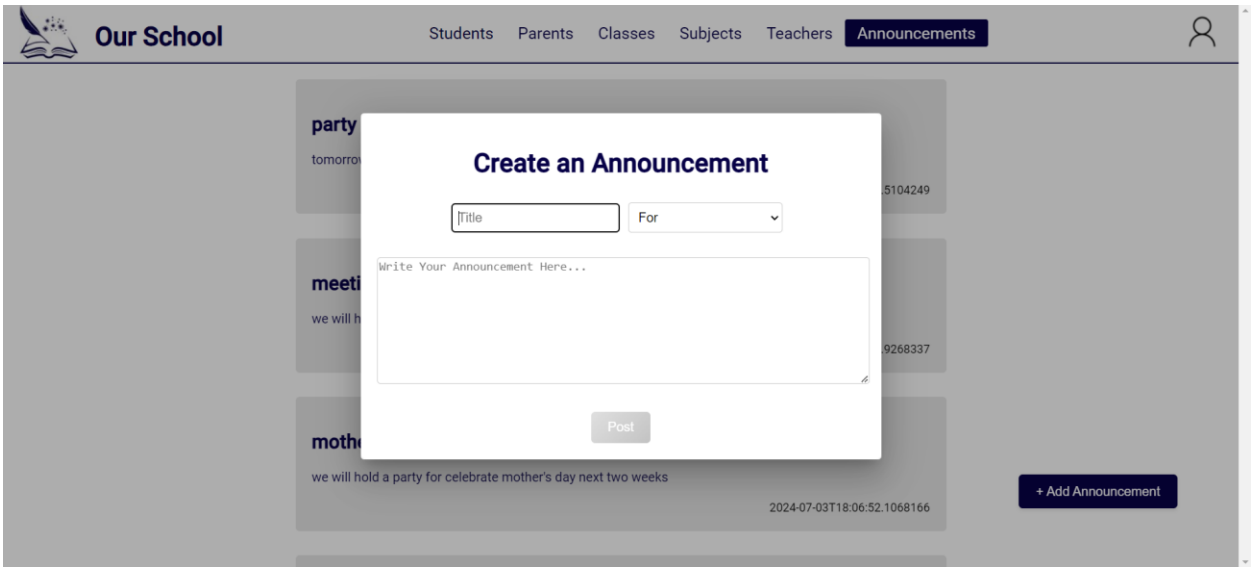


Figure 19: Add Announcement

Add Evaluation Report

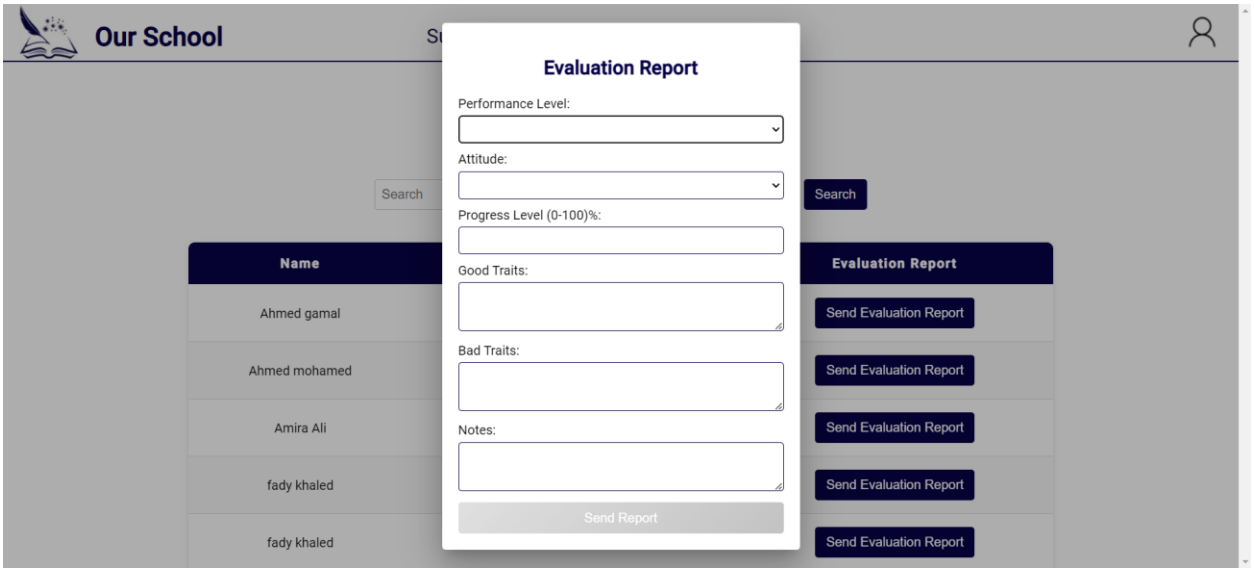


Figure 20: Add Evaluation Report

Chapter 5: Implementation and Testing

5.1. Implementation

Frontend

- Angular

  Our main technology for the front-end of our website was Angular, we chose Angular because it offers the following benefits:

  - Improved Speed and Performance
  - Easy and effective communication with a RESTful server
  - High quality of Angular applications
  - Easy to transition to cross-platform development using ionic

Backend

- C#
- ASP. Net Core
- SQL server

Machine Learning

5.2. Testing

5.2.1. Back-End

1. Login User

Test Scenario	*Steps*	*Expected Result*	*Status*
1. User login with valid data	1. Enter email. 2. Enter password.	User logged in successfully	Passed
2.User login with wrong email	1. Enter wrong email. 2. Enter wrong password.	Error Message: "Email or Password is incorrect." Login failed.	Passed
3. User login with wrong password	1. Enter email. 2. Enter wrong password.	Error Message: "Email or Password is incorrect." Login failed.	Passed

Table 6: Test Table 1 — Login User

1.1. User Login with Valid Data

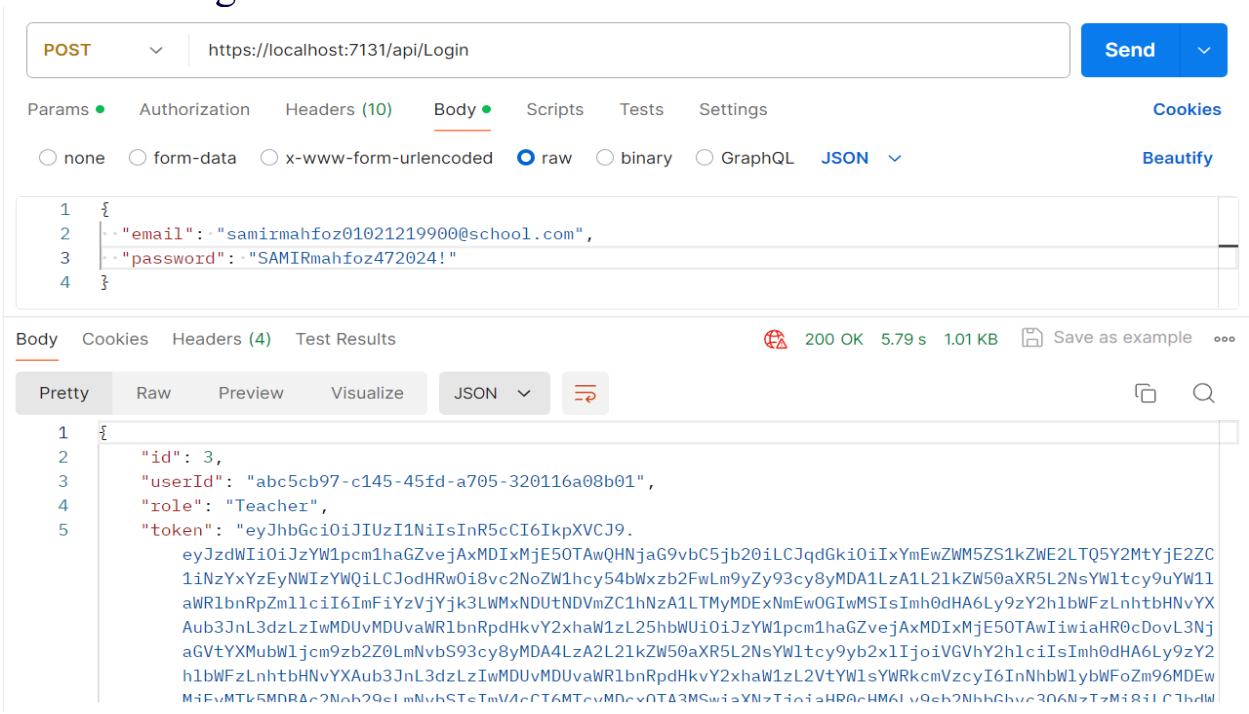


Figure 21: User Login with Valid Data

1.2. User login with wrong email

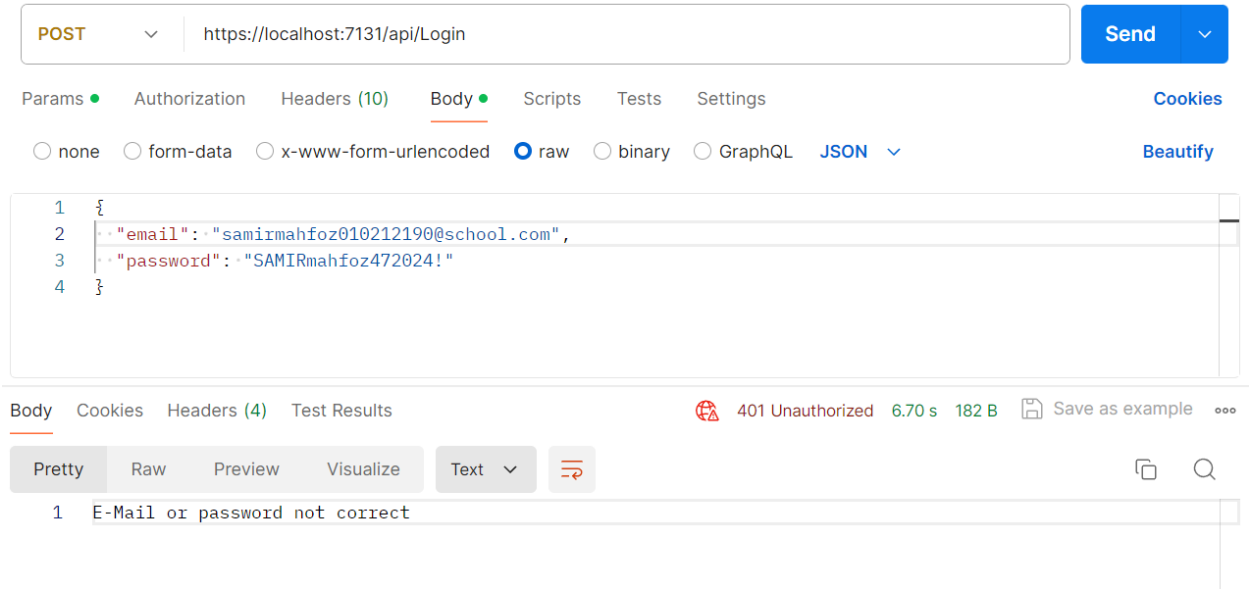


Figure 22: User login with wrong email

1.3. User login with wrong password

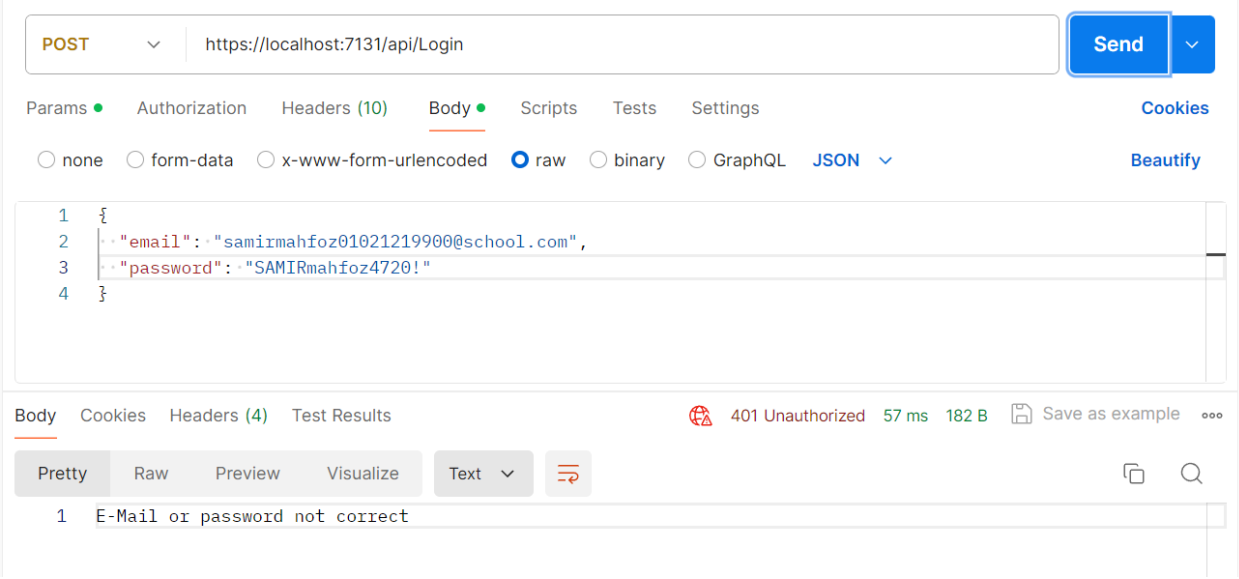


Figure 23: User login with wrong password

2. Add Parent

Test Scenario	*Steps*	*Expected Result*	*Status*
1. Add Parent with valid data.	1. Enter name. 2. Enter Gmail. 3. Enter phone number. 4. Enter address.	Parent added successfully.	Passed
2. Add Parent with first name only.	1. Enter first name. 2. Enter Gmail. 3. Enter phone number. 4. Enter address.	Error message: "name must be at least first and second." Added failed.	Passed
3. Add Parent with invalid Gmail.	1. Enter name. 2. Enter invalid Gmail. 3. Enter phone number. 4. Enter address.	Error message: "this is invalid Gmail address it must terminate with @gmail.com." Added failed.	Passed
4. Add Parent with phone number less than 11 number.	1. Enter name. 2. Enter Gmail. 3. Enter phone number less than 11 number. 4. Enter address.	Error Message: "phone number must be 11 number." Added failed	Passed
5. Add Parent with invalid phone number.	1. Enter name. 2. Enter Gmail. 3. Enter invalid phone number. 4. Enter address.	Error Message: "Your phone number must begin with 011, 010, 012, or 015." Added failed	Passed

Table 7: Test Table 2 — Add Parent

2.1 Add Parent with valid data

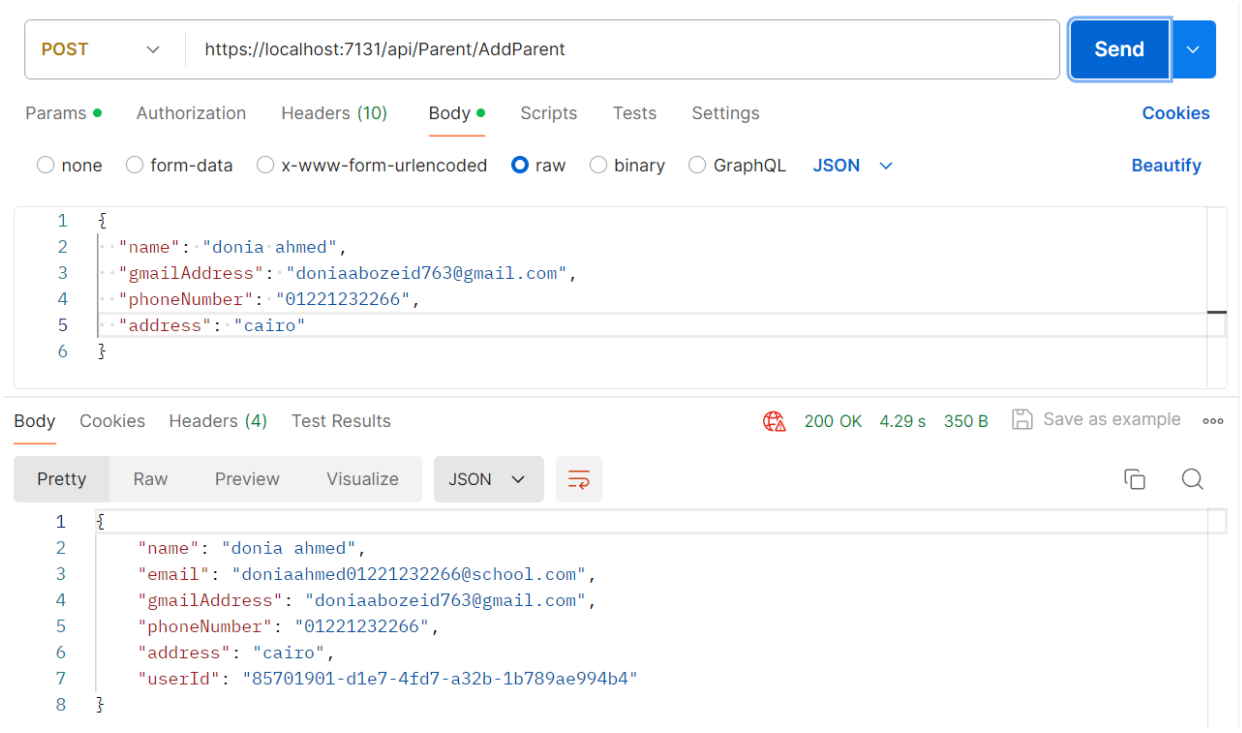


Figure 24: Add Parent with valid data

2.2 Add Parent with first name only

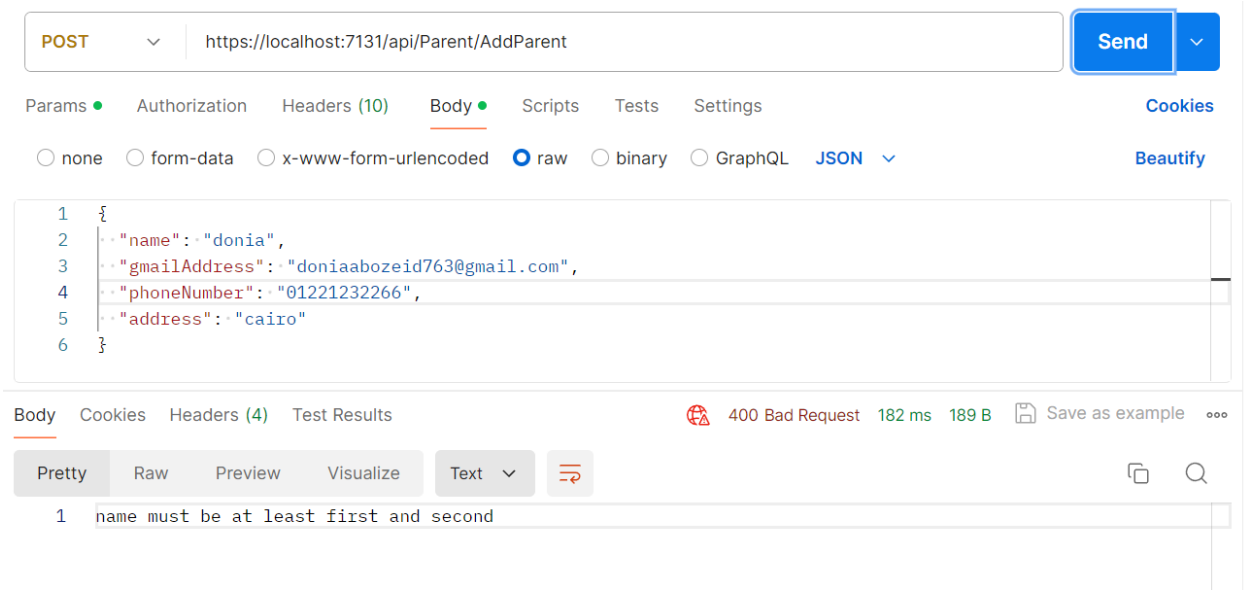


Figure 25: Add Parent with first name only

2.3 Add Parent with invalid Gmail

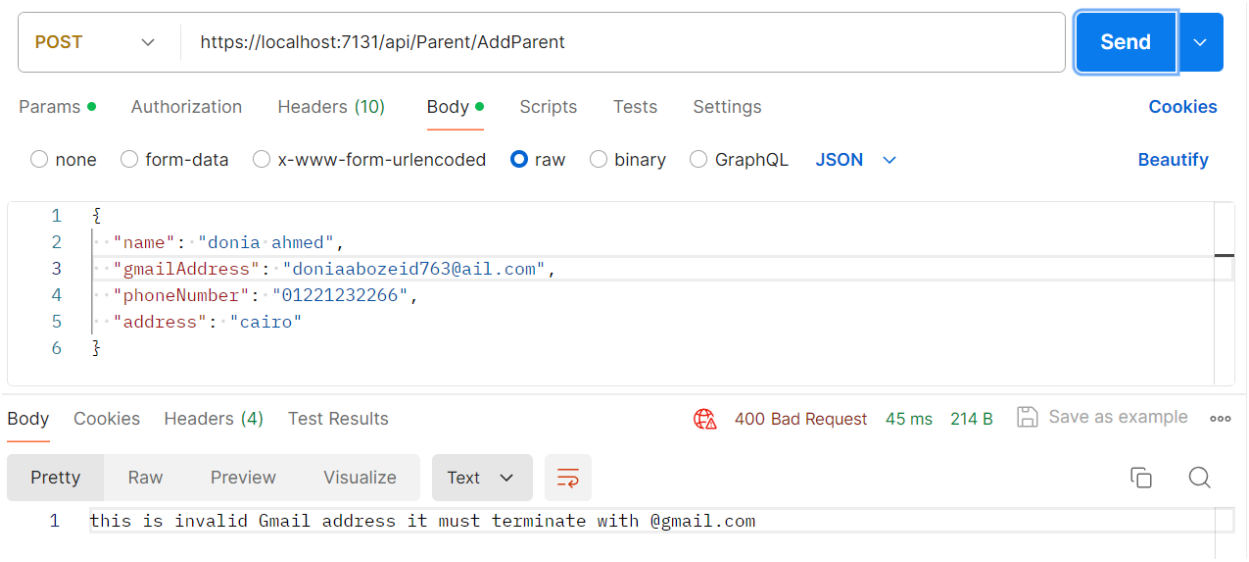


Figure 26: Add Parent with invalid Gmail

2.4 Add Parent with phone number less than 11 numbers

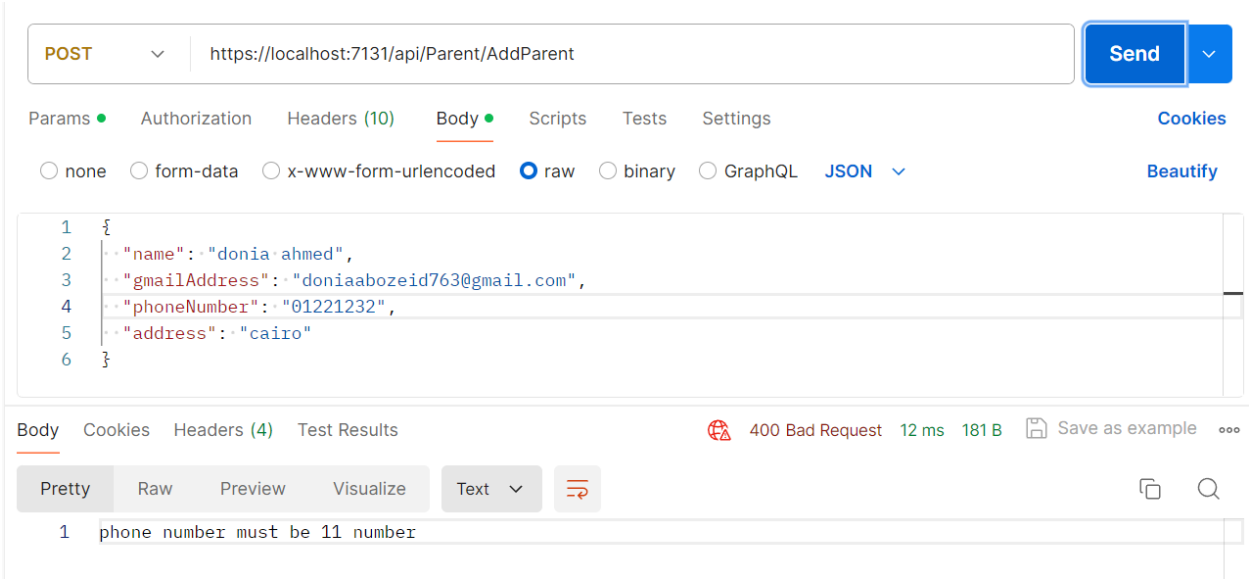


Figure 27: Add Parent with phone number less than 11 numbers

2.5 Add Parent with phone number less than 11 numbers

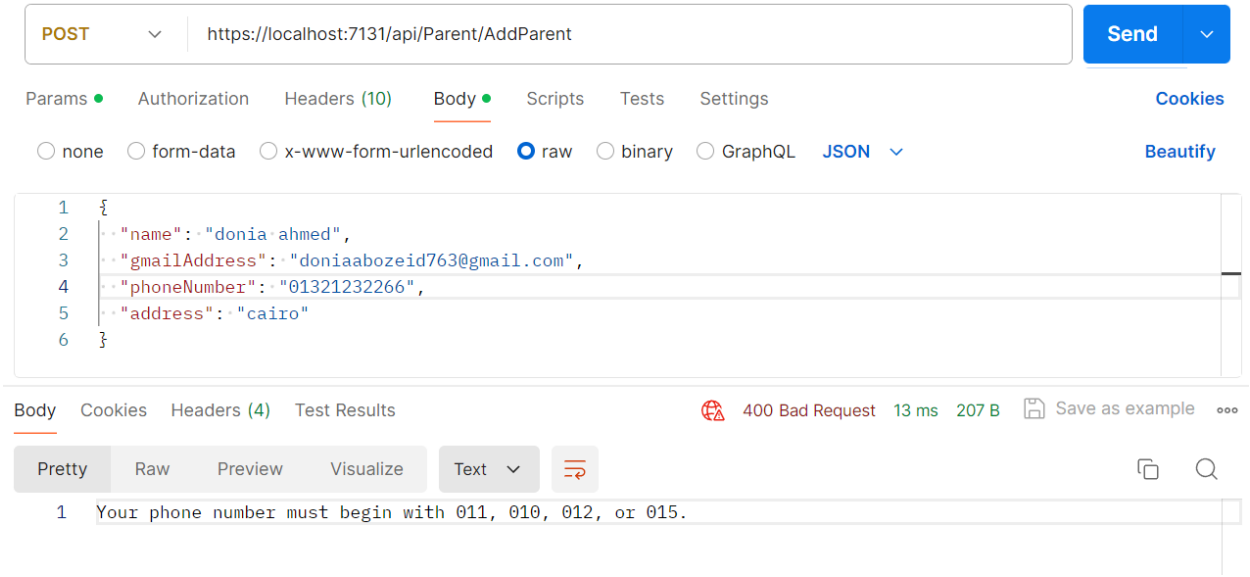


Figure 28: Add Parent with phone number less than 11 numbers

3. Send Announcement

Test Scenario	*Steps*	*Expected Result*	*Status*
1. Admin send a valid Announcement.	1. Enter title. 2. Enter Body. 3. Enter this Announcement for which.	Announcement added successfully.	Passed
2. Admin send an Announcement with empty title.	1. Enter empty title. 2. Enter Body. 3. Enter this Announcement for which.	Error Message: " Title mustn't be empty." sending failed.	Passed
3. Admin send an Announcement with empty Body.	1. Enter title. 2. Enter empty Body. 3. Enter this Announcement for which.	Error Message: "Message mustn't be empty." sending failed.	Passed
4. Admin send an Announcement with invalid number for -> for which.	1. Enter title. 2. Enter Body. 3. Enter invalid number for Announcement for which.	Error Message: " you must choose 1 for Student or 2 for Parent or 3 for Teacher or 4 for All of them." sending failed.	Passed

Table 8: Test Table 3 — Send Announcement

3.1. Admin Sends a valid Announcement

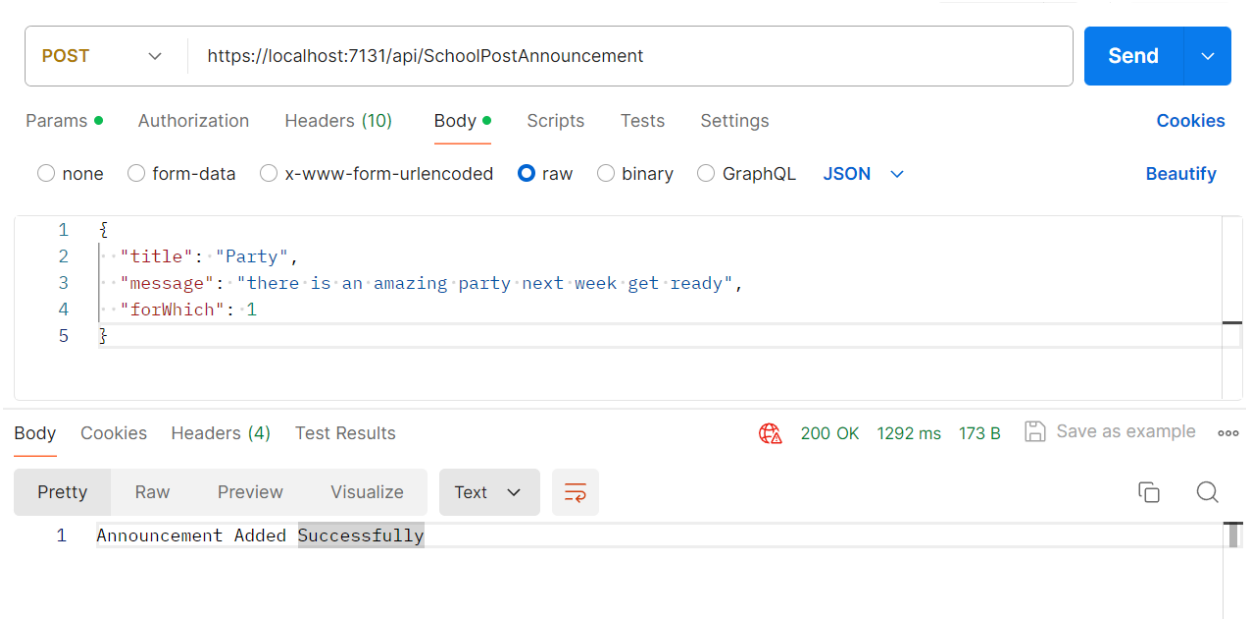


Figure 29: Admin Sends a valid Announcement

3.2. Admin sends an Announcement with empty title

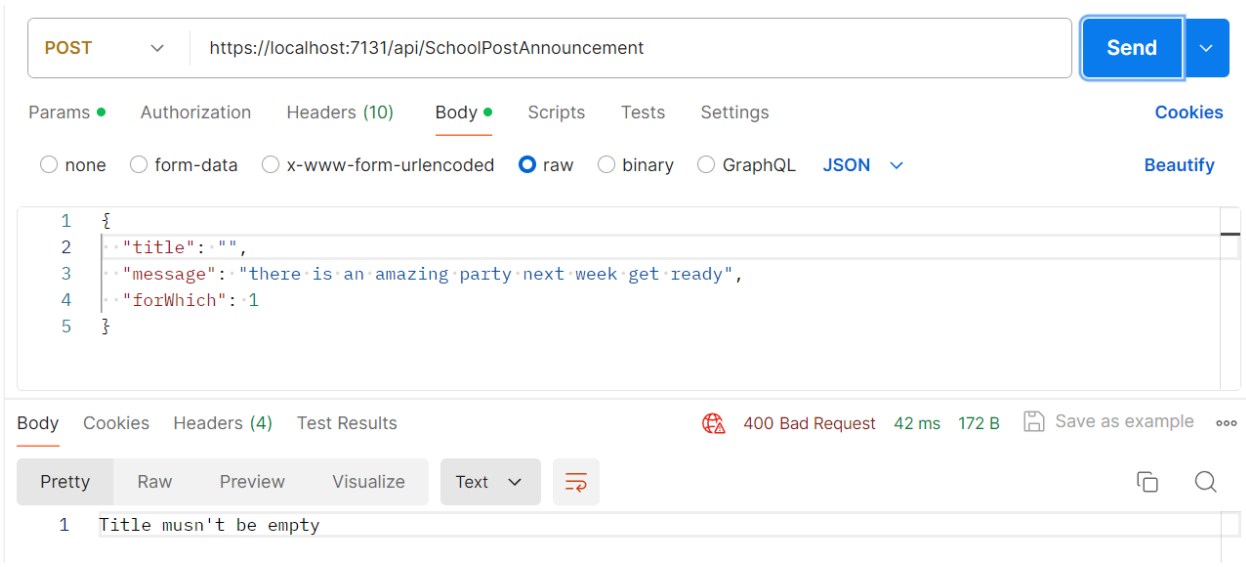


Figure 30: Admin sends an Announcement with empty title

3.3. Admin sends an Announcement with empty Body

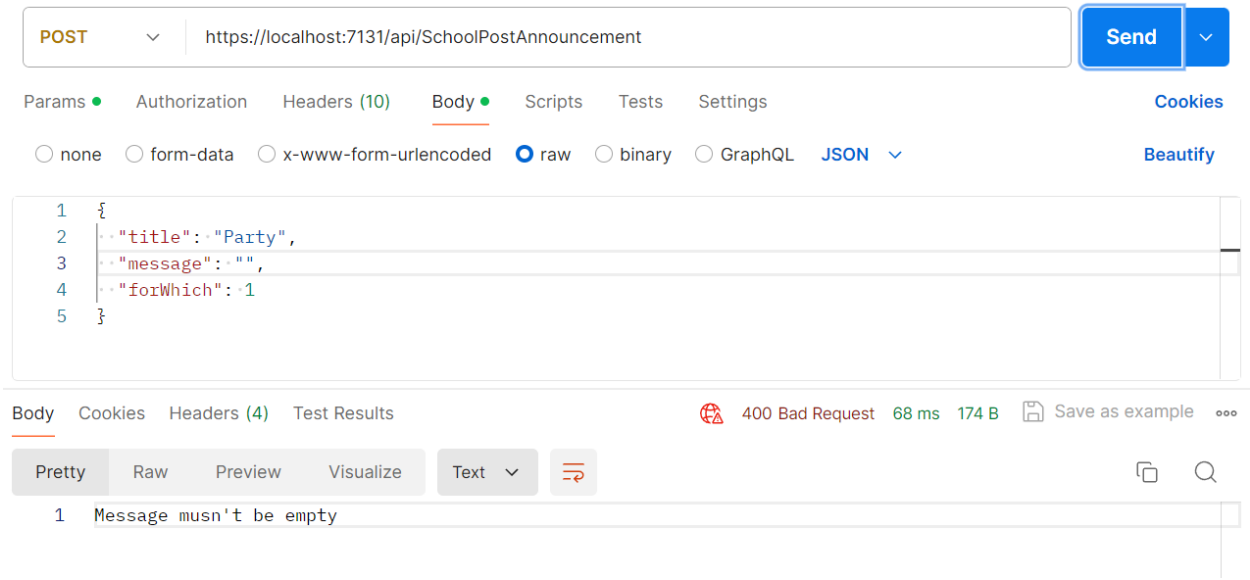


Figure 31: Admin sends an Announcement with empty Body

3.4. Admin sends an Announcement with invalid number for -> for which

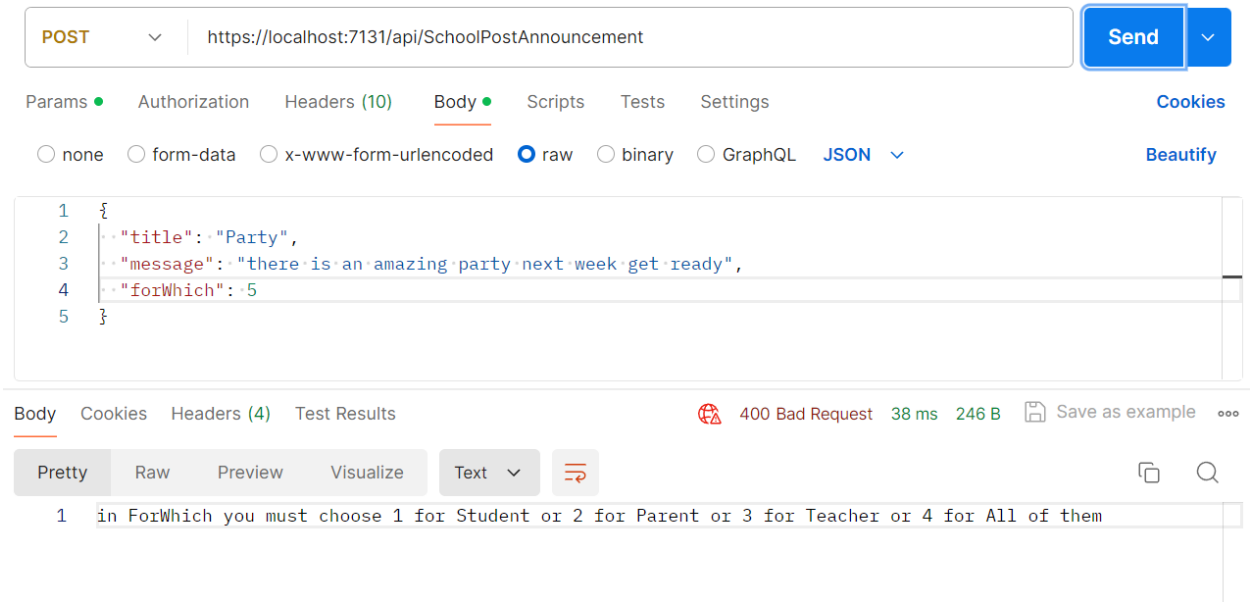


Figure 32: Admin sends an Announcement with invalid number

4. Add Class

Test Scenario	*Steps*	*Expected Result*	*Status*
1. Admin choose level and department and enter number of Class that not Exist in level	1. Choose Level. 2. Choose Department. 3. Enter Number of Class.	Class added successfully.	Passed
2. Admin choose level and department and enter number of Class that Already Exist in level	1. Choose Level. 2. Choose Department. 3. Enter Number of Class Already Exist.	Error Message: Number Of Class in This Leve Already Exist " Adding failed.	Passed

Table 9: Test Table 4 — Add Class

4.1. Admin chooses level and department and enter number of Class that not Exist in level

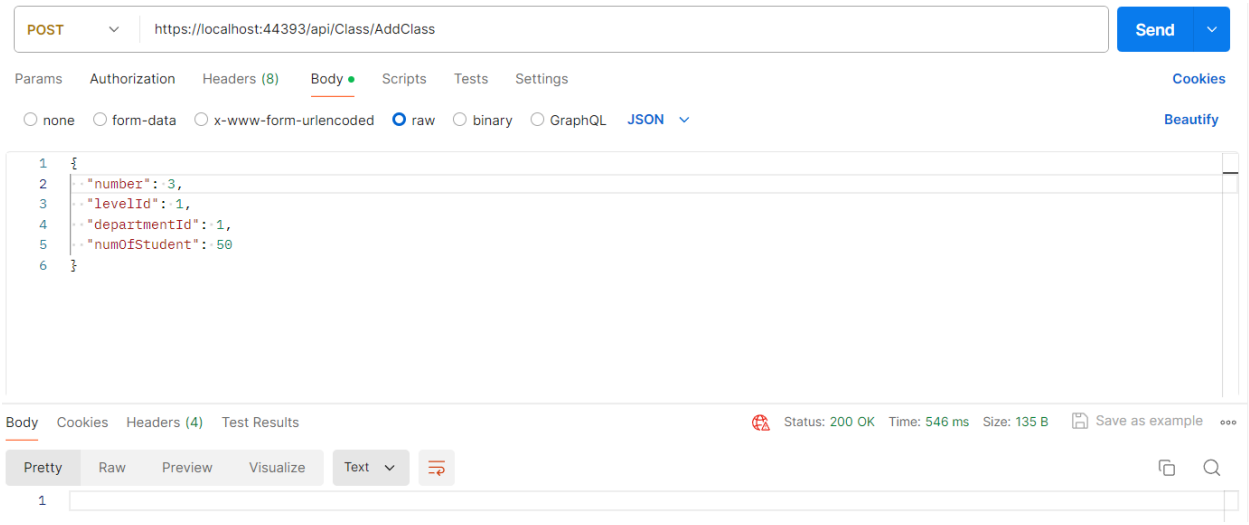


Figure 33: Admin chooses dept and enter no. of Class that not Exist in level

4.2. Admin chooses level and department and enter number of Class that Already Exist in level

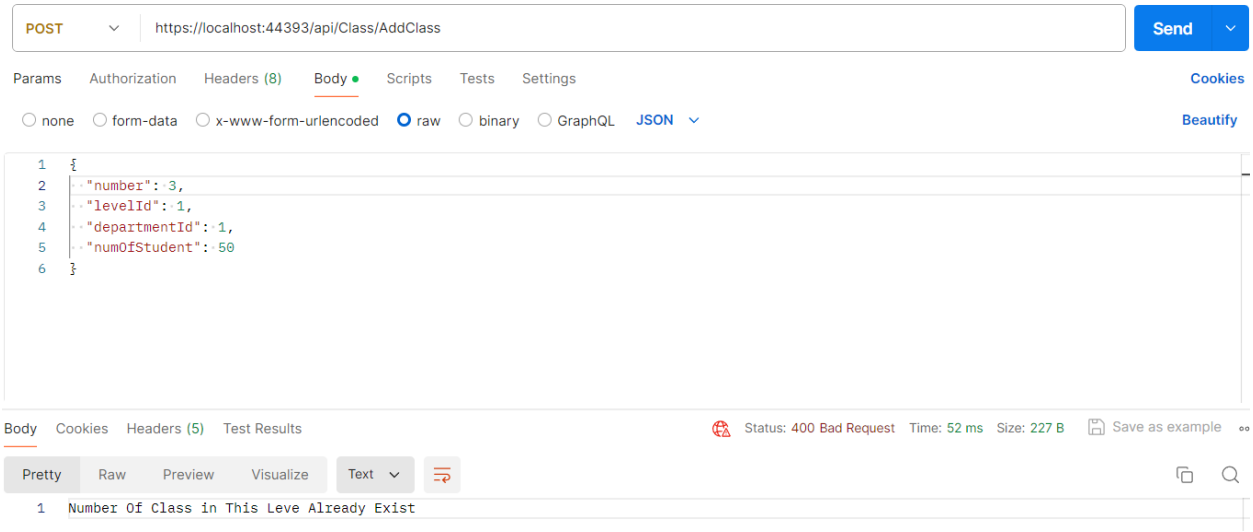


Figure 34: Admin chooses dept and enter no. of Class that Already Exist in level

5. Update Final Degree of Subjects for Student

Test Scenario	*Steps*	*Expected Result*	*Status*
1. Admin Enter Valid Degrees for Subjects	1. Determine student. 2. Enter valid Grades that with a range 0 to final degree Limit. 3.Press Save	Grades saved successfully.	Passed
2. Admin Enter invalid Degrees for Subjects.	1. Determine student. 2. Enter invalid Grades for Subject (less than 0 or greater than final degree Limit). 3.Press Save	Error Message: "Final Degree for subject must be greater than or equal zero and less than or equal “Limit final Degree”." Update Degree failed.	Passed

Table 10: Test Table 5 — Update Final Degree of Subjects for Student

5.1. Admin Enters Valid Degrees for Subjects

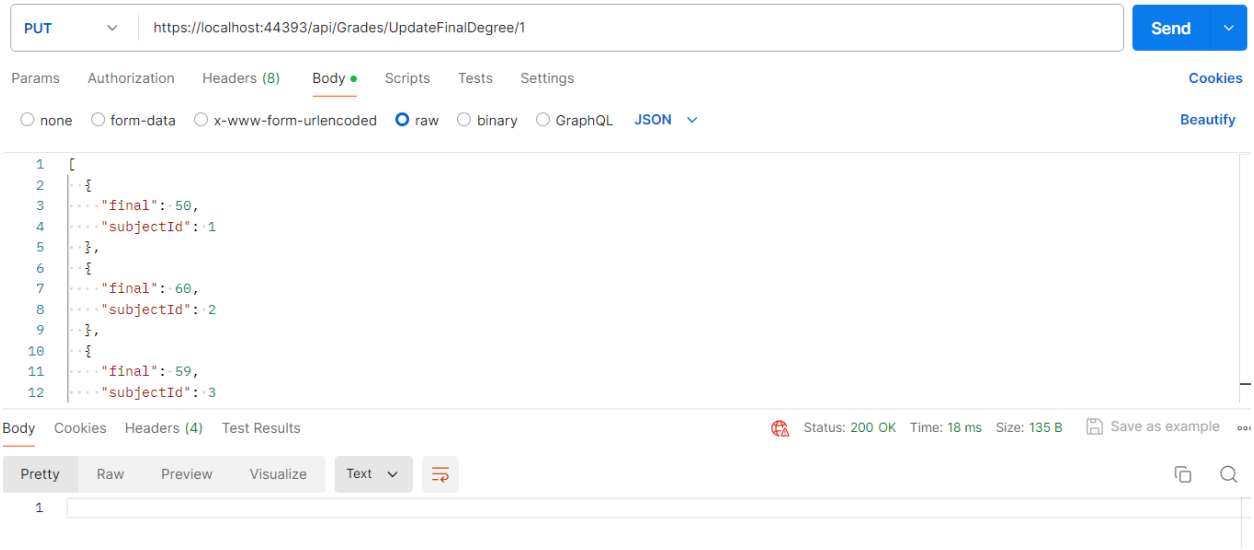


Figure 35: Admin Enters Valid Degrees for Subjects

5.2. Admin Enters Invalid Degrees for Subjects

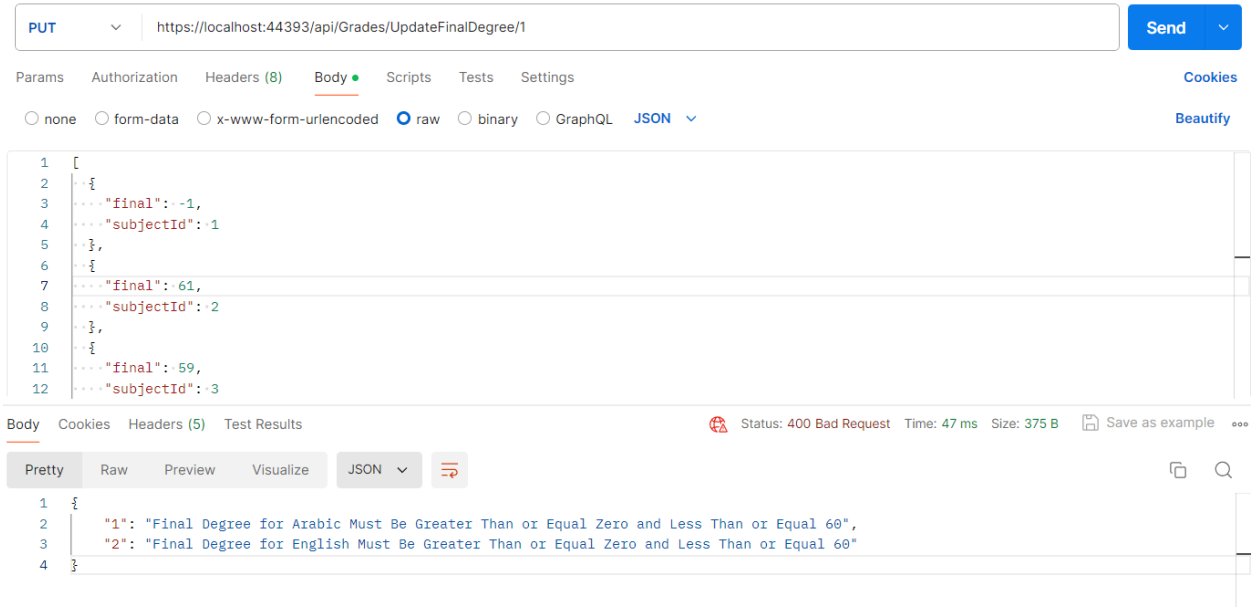


Figure 36: Admin Enters Invalid Degrees for Subjects

6. Add Subject Record

Test Scenario	*Steps*	*Expected Result*	*Status*
1. Admin Enter Valid Degrees for Subjects	1. Determine student. 2. Enter valid Grades that with a range 0 to final degree Limit. 3.Press Save	Grades saved successfully.	Passed
2. Admin Enter invalid Degrees for Subjects.	1. Determine student. 2. Enter invalid Grades for Subject (less than 0 or greater than final degree Limit). 3.Press Save	Error Message: "Final Degree for subject must be greater than or equal zero and less than or equal “Limit final Degree”." Update Degree failed.	Passed

Table 11: Test Table 6 — Add Subject Record

6.1. Admin Enters Subject Record that wasn't repeated

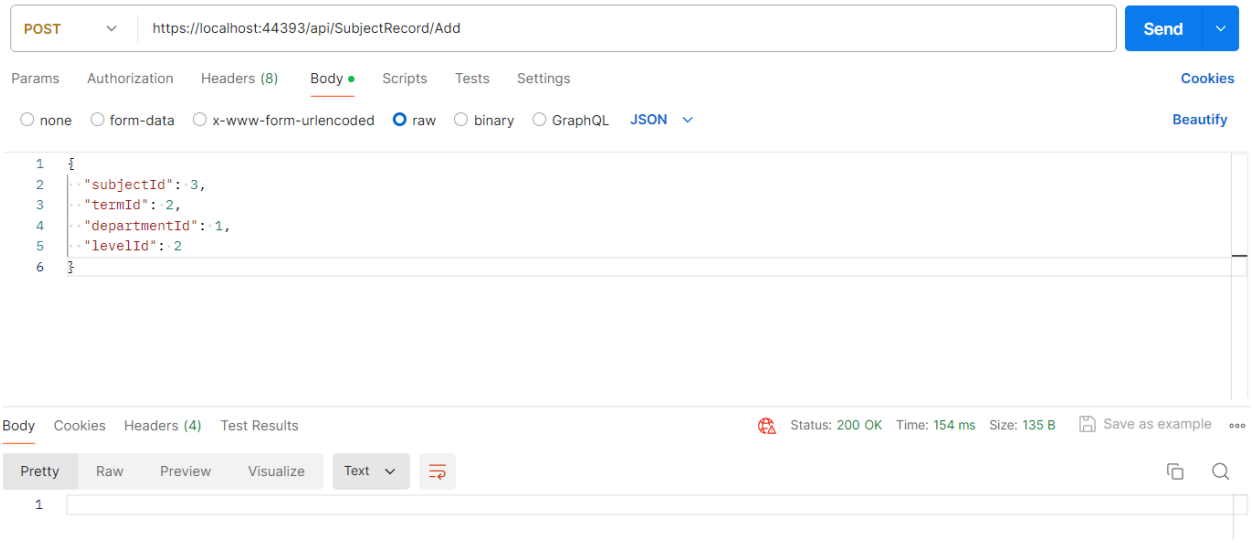


Figure 37: Admin Enters Subject Record that wasn't repeated

6.2. Admin Enters Subject Record that was repeated

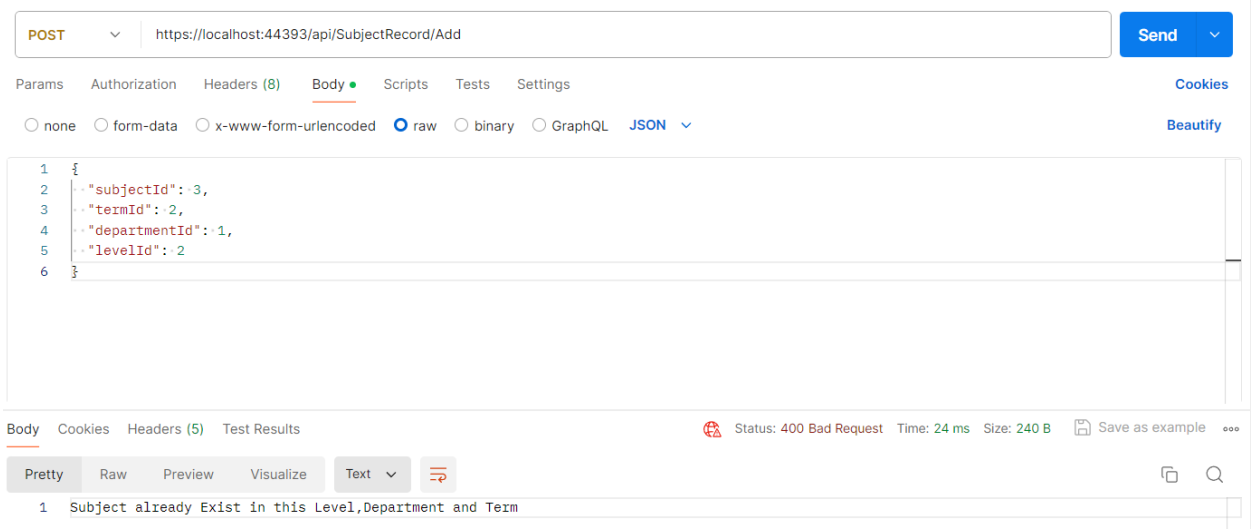


Figure 38: Admin Enters Subject Record that was repeated

5.2.2. Front-End

Wrong Password

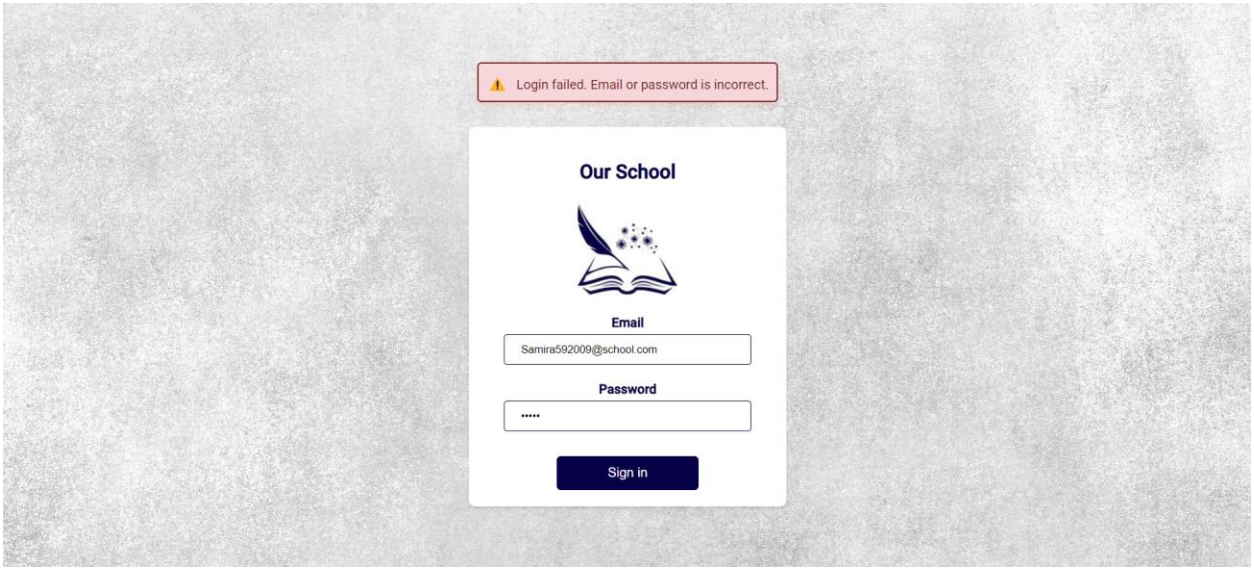


Figure 39: Wrong Password

Wrong Email

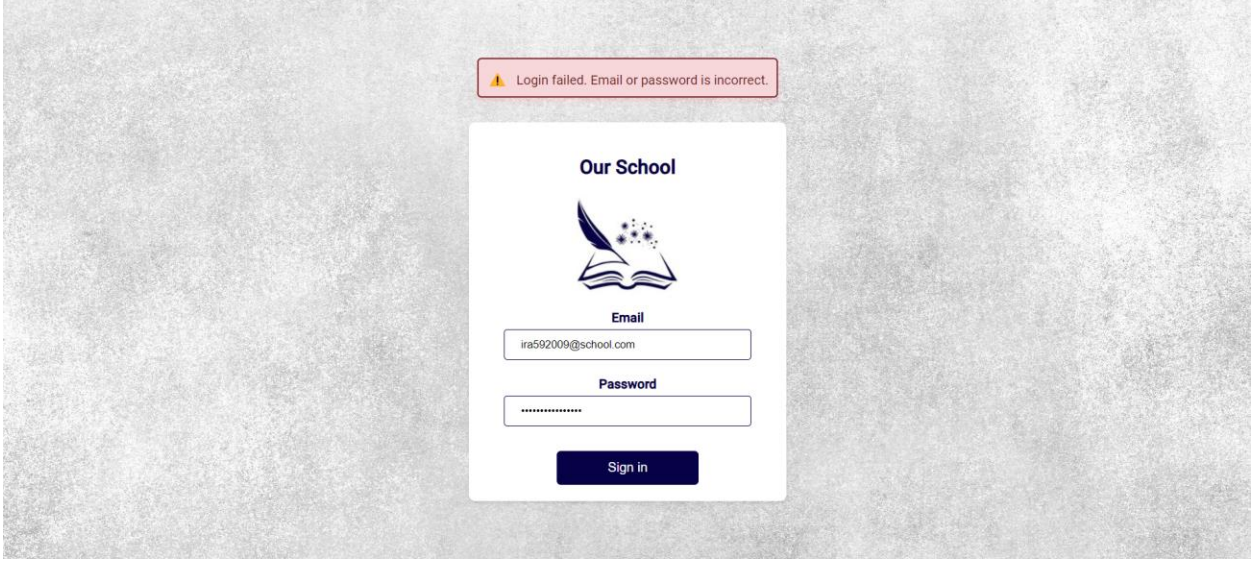


Figure 40: Wrong Email

Parent Validation

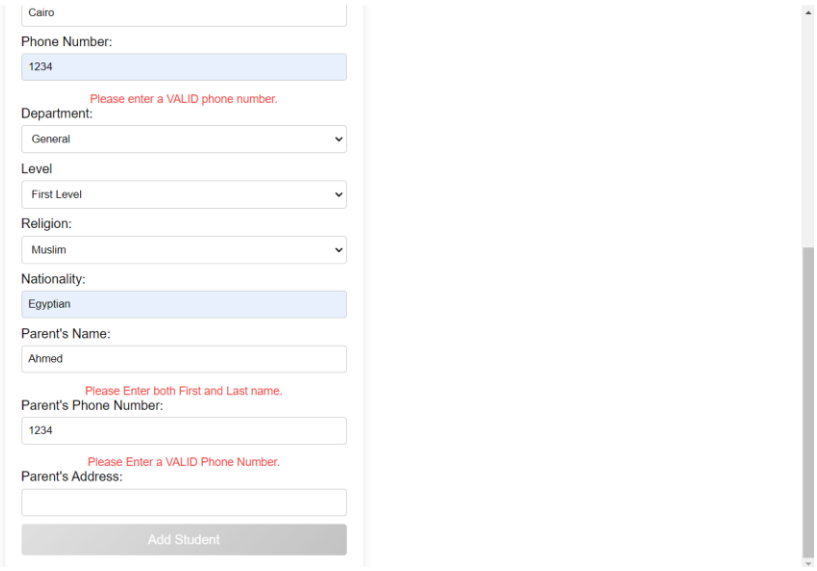


Figure 41: Parent Validation

Parent Successfully Added

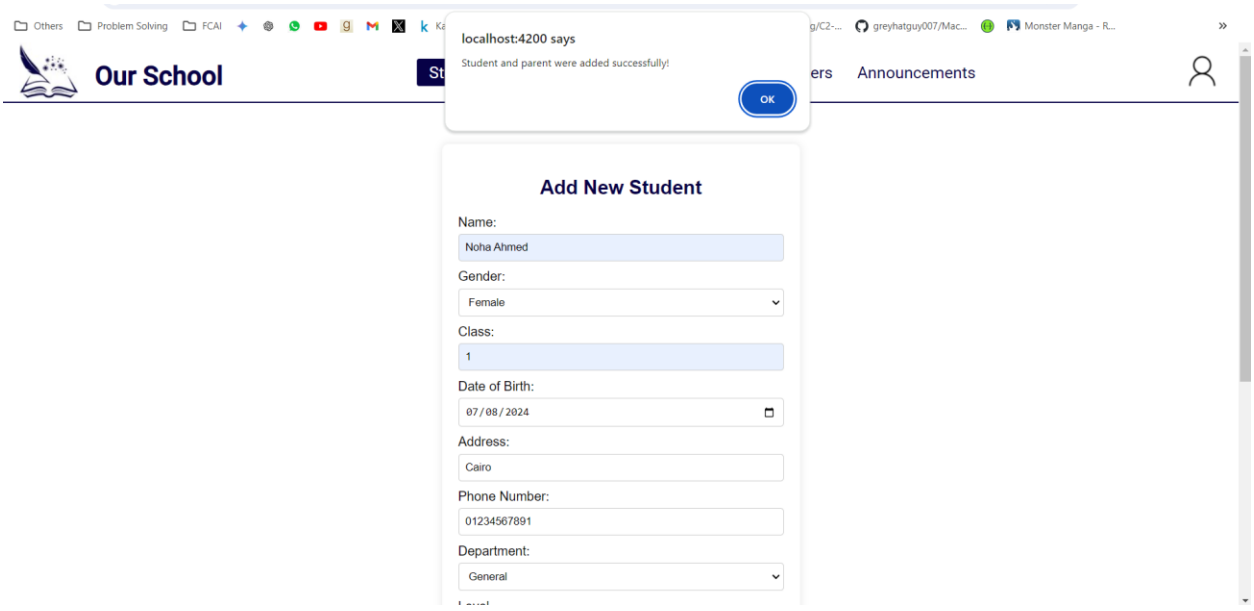


Figure 42: Parent Successfully Added

Announcement Validation

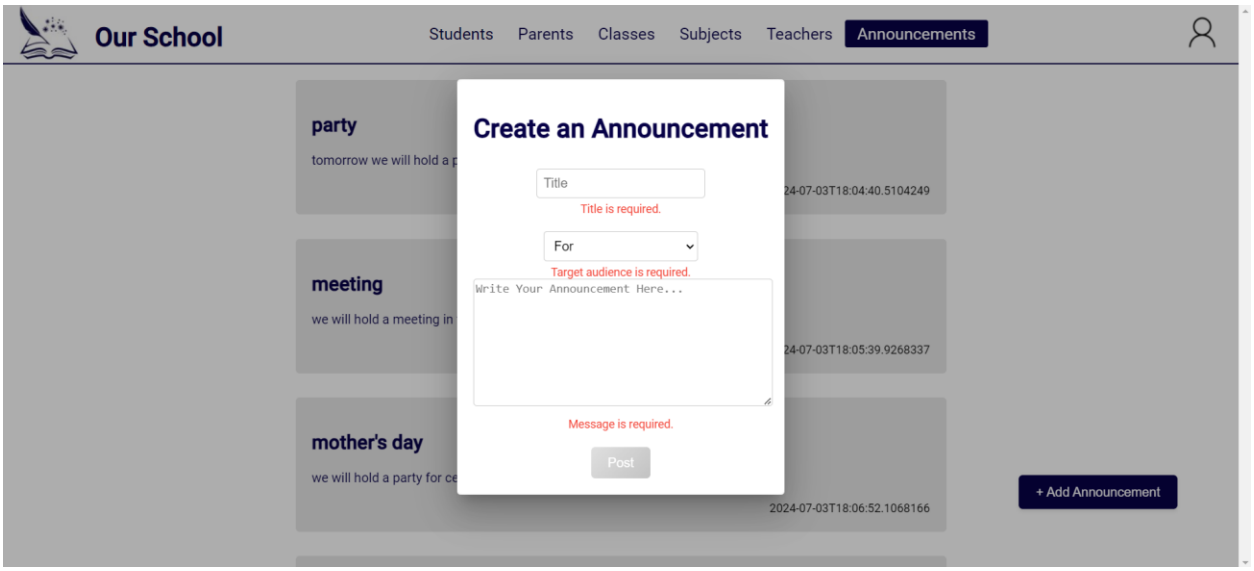


Figure 43: Announcement Validation

Announcement Successfully Post

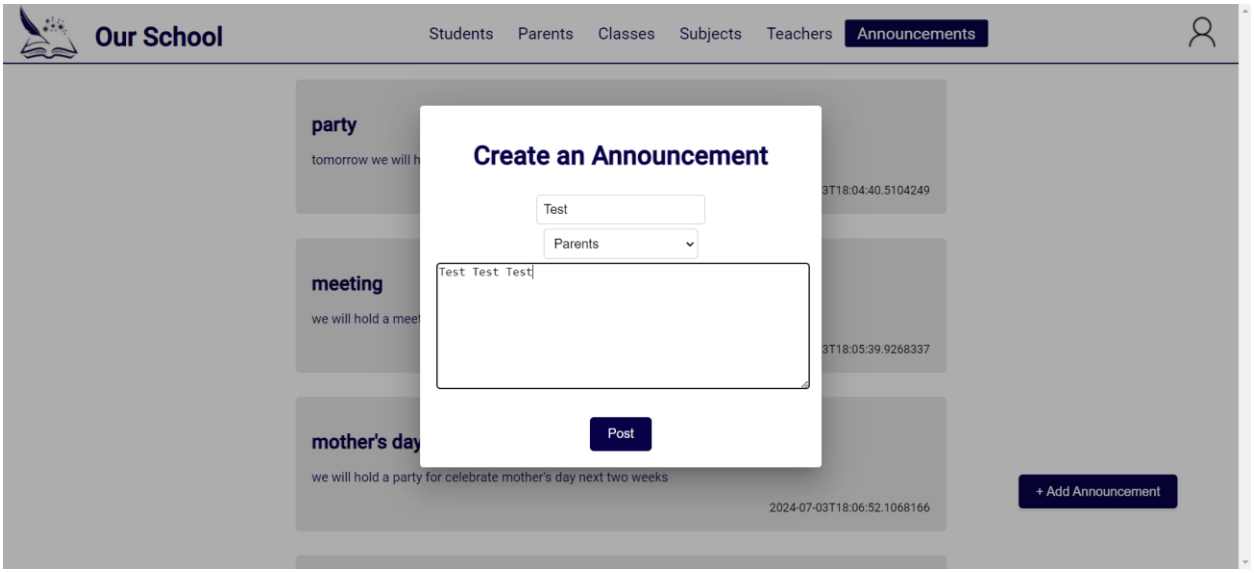


Figure 44: Announcement Successfully Post

References

[1] *Angular—Introduction to the Angular docs*. (n.d.). Retrieved February 10, 2024, from https://angular.io/docs

[2] *ASP.NET core tutorial for beginners—YouTube*. (28/12/2018). Retrieved February 10, 2024, from https://www.youtube.com/playlist?list=PL6n9fhu94yhVkdrusLaQsfERmL_Jh4XmU

[3] BillWagner. (n.d.). *NET documentation*. Retrieved February 10, 2024, from https://learn.microsoft.com/en-us/dotnet/

[4] *Google Classroom*. (n.d.). Retrieved February 10, 2024, from https://sites.google.com/view/classroom-workspace/

[5] Schwarzmüller, M. (1/1/2024). *Master Angular 7 (formerly Angular 2): The Complete Course*. Udemy. Retrieved February 10, 2024, from https://www.udemy.com/course/the-complete-guide-to-angular-2/

[6] *SQL Server tutorial for beginners—YouTube*. (4/8/2012). Retrieved February 10, 2024, from https://www.youtube.com/playlist?list=PL08903FB7ACA1C2FB

[7] *The starting point for learning TypeScript*. (n.d.). Retrieved February 10, 2024, from https://www.typescriptlang.org/docs/

[8] *What is PowerSchool?* (n.d.). PowerSchool. Retrieved February 10, 2024, from https://www.powerschool.com/global/africa-get-started/