



National University of Sciences &
Technology (NUST), ISLAMABAD

CS220 Database Systems BESE 2K21-B

END SEMESTER PROJECT REPORT

Group Members:

SADDAM HUSSAIN 370126
YUVRAJ SAGAR 373712
M. TALHA HABIB 369654

BESE-12-B

Submitted To:

Dr. BILAL ALI RIZVI

School of Electrical engineering and Computer Sciences
(SEECs),
NUST.

Contents

Abstract:	4
Scope:	4
Objectives:	4
Purpose:	4
Literature Review:	5
Requirements Analysis:	5
Functional Requirements:	5
Non-Functional Requirements:	5
Expected Users:	6
Functional Requirements for Admin:	6
View the Academic Activities:	6
Manage Student:	6
Manage Staff:	6
Manage Classes:	6
Manage Subjects:	6
Manage Session Years:	6
View Attendance:	7
Student and Staff Feedback:	7
Student and Staff Leave:	7
Functional Requirements for Staff:	7
Check Students Under him:	7
Manage Attendance:	7
Add Result:	7
Apply Leave:	8
Give Feedback:	8
Functional Requirements for Students:	8
View Subjects Enrolled:	8
View Attendance:	8
View Result:	8
Apply For Leave:	8
Give Feedback:	8
Non-Functional Requirements:	9

Security Requirements:.....	9
User-Friendly Interface:	9
System Requirements:	9
Design:	9
Types of designs:.....	9
Architectural design:.....	9
Use Case Diagram:	10
Database design:.....	10
ERD Diagram:	11
User interface design:	12
Testing:.....	17
Conclusion:.....	19

Academy Management System

Abstract:

The purpose of this project is to develop a web-based academy management system that will streamline and automate various administrative tasks, such as student enrollment, course scheduling, and grading. The system will be designed to improve efficiency and productivity for academy staff, as well as provide a user-friendly interface for students to access their academic records and course materials.

Scope:

The academy management system will cover the following functionalities:

- Student enrollment
- Course scheduling
- Grading and assessment
- Student records
- Feedback management
- Communication between students, teachers, and staff
- Attendance management
- Apply for leave

The system will be implemented for a medium-sized academy with a diverse student body and a range of academic programs. It will be developed using web technologies and will be accessible via a web browser.

Objectives:

The main objectives of this project are to:

- Improve the efficiency and effectiveness of academy management processes
- Provide a user-friendly interface for students to access their academic records and course materials
- Enhance communication and collaboration among students, teachers, and staff
- Reduce the reliance on paper-based systems and manual processes
- Increase data security and privacy

Purpose:

The web-based academy management system will serve as the central hub for all academy-related activities, providing a single source of truth for student and course information. It will enable academy staff to manage and track student progress, as well as facilitate communication and collaboration between students and teachers. By streamlining and automating administrative tasks,

the system will free up time and resources for more important tasks, such as teaching and learning. Ultimately, the goal of this project is to improve the quality of education and the overall student experience at the academy.

Literature Review:

The literature review for an academy management system should review relevant research and literature on topics such as:

- **Existing academy management systems:** This could include descriptions and evaluations of different software solutions that are currently used for academy management.
- **Best practices for academy management:** This could include research on effective strategies and approaches for managing academies, such as student enrollment, course scheduling, and grading.
- **User needs and requirements:** This could include research on the specific challenges and needs of academy staff and students in terms of managing and accessing academic information.
- **Standards and regulations:** This could include research on any relevant industry standards or regulations that may impact the design and development of the academy management system.
- **Technologies and tools:** This could include research on web technologies and other tools that may be relevant for the development and deployment of the academy management system.
- **Case studies:** This could include research on real-world examples of academies that have successfully implemented academy management systems and the benefits they have experienced.

Overall, the literature review should provide a comprehensive overview of the current state of the field and highlight any gaps or opportunities for improvement in the area of academy management. It should also provide a strong foundation for the requirements analysis and design of the academy management system.

Requirements Analysis:

Functional Requirements:

The Functional requirements describe the specific actions or behaviors that a system should be able to perform. These requirements specify what the system should do, and are typically expressed in terms of inputs, outputs, and processes.

Non-Functional Requirements:

Non-functional requirements, on the other hand, describe the overall characteristics and qualities that a system should possess. These requirements specify how the system should behave, and are typically expressed in terms of performance, reliability, security, and other factors.

Expected Users:

We have **three** Expected Users for our Academy Management System i.e., The Admin, the Staff, and the Student. There have been different Functional Requirements for the Actors in the Project. These are being discussed below:

Functional Requirements for Admin:

View the Academic Activities:

Our system provides a clear and informative visualizations of the overall activities at the campus which we have implemented using graphs and different navbars. The admin can monitor these activities and can take actions in case something becomes extraneous.

Manage Student:

The system allows administrators to easily enter and update student information, such as contact details, enrollment status, and academic records. Also, he can add new students and delete students too. During the Insertion, proper Validation is done to abstain from the Data Clash and to ensure the accuracy. The privacy and data integrity is also managed as we have assigned a proper Login username and password to restrain the unauthorized access or tampering.

Manage Staff:

Like Manage Students, the admin can add and delete staff too. The above-mentioned constraints and checks are also implemented for the Insertion of Staff too.

Manage Classes:

The admin can add, update, and delete the classes which are being taught at their Academy. The input constraints are properly managed to avoid incorrect data insertion.

Manage Subjects:

The admin can add, update, and delete the subjects which they are offering at their Academy. The input constraints are properly managed to avoid incorrect data insertion and maintain data integrity.

Manage Session Years:

The admin can add, update, and delete the session years information such as start and end dates. The session years are basically the period for which they are offering subjects at their Academy. The session years decides that which batch is being in Progress and makes the data filtration for

the Past and in Progress students. The input constraints are properly managed to avoid incorrect data insertion.

View Attendance:

The admin can look out the attendance of the students in different subjects of different classes. He'll select the subject of a class to see the attendance of the students over a range of days he/she enters. Then he/she can draw the concluding remarks about these attendance results.

Student and Staff Feedback:

The admin can solve out the different queries of the student and the staff using this Feedback attribute. It is basically a comment of a student or a staff on something he/she wants to tell the admin. It can include the complain about any event occurring at academy or any other reason.

Student and Staff Leave:

The admin can overlook the student and staff Leaves. He will manage these leaves and can manage their records of Leaves. He will approve or Reject the Leave request from students and Staffs according to the Leave Reason they have provided.

Functional Requirements for Staff:

Check Students Under him:

The staff can check out the total student under him in a specific subject. From, here

This requirement allows staff members to view the list of students that are current students of him/her. This includes data such as the student's name, class, enrollment status, and academic progress.

Manage Attendance:

This specifies that the staff members can update attendance records for the students under their care. This includes marking students as present or absent, and recording any excused or unexcused absences. Further, these attendances are also available to students and Admin.

Add Result:

The staff members can enter and update the academic results of the students under their care, such as test scores. The system validates data input to ensure that it is accurate and complete. These Results are further accessible to Students and passively to Admin too.

Apply Leave:

The members can request for a leave of absence, such as for vacation or personal time. The system supports the submission and approval of leave requests and should track the remaining leave balance for each staff member. These leave requests are handled by the Admin. He'll approve or reject the request on basis of the Leave Reason.

Give Feedback:

The staff members can provide feedback on various aspects of the academy, such as teaching quality, facilities, or student performance. The system should support the submission and review of feedback and should allow staff members to communicate their positive and critical thoughts with Admin in order take decision for the betterment of Academy.

Functional Requirements for Students:

View Subjects Enrolled:

The students can view the list of subjects that they are enrolled in. This includes details such as the subject name, and instructor etc.

View Attendance:

The students can view their own attendance records. This includes data such as the dates of attendance, the percentage of classes attended, and any excused or unexcused absences. From this, students are alarmed of their absences to prevent the loss.

View Result:

The students can view their own academic results, such as grades or test scores. This includes data such as the student's overall scores on specific exams and quizzes.

Apply For Leave:

The students can request a leave of absence, such as for illness or personal reasons. The system supports the submission and approval of leave requests and should track the remaining leave balance for each student. Like staff leaves, these leave requests are handled by the Admin. He'll approve or reject the request on basis of the Leave Reason.

Give Feedback:

Like Staff Feedback, the students can provide feedback on various aspects of the academy, such as teaching quality, facilities, or academic support, too. The system supports the submission of feedback and allows students to communicate their positive and critical thoughts with Admin.

Non-Functional Requirements:

Security Requirements:

These requirements pertain to the measures that need to be in place to ensure the security and confidentiality of the data stored in our Academy Management system. This includes confidentiality measures such as secure login protocols, data encryption, and access controls. Also, during the data insertion in the Database, we have encrypted the sensitive information like login password using SHA-256 Hash Algorithm.

User-Friendly Interface:

This requirement pertains to the design and usability of the system and refers to the need for the system to be easy for users to navigate and use. A user-friendly interface can improve user satisfaction and productivity. We have managed to create a very easy to use User-Interface.

System Requirements:

These requirements describe the hardware and software components that are necessary for the system to function correctly. This includes specifications for the server, operating system, database, and any other software or hardware components that are required. Our project is quite manageable and used on Windows. We haven't deployed the project Online. It is purely web-based but for now we are executing on localhost:8000.

Design:

Types of designs:

There are many types of designs that can be used in the development of a software project. Some major types of designs which will we discuss for our YUVI Academy project:

Architectural design:

This type of design defines the overall structure of the software system, including the relationships between different components and the interfaces between those components. An architectural design for our academy management system includes the following components and relationships:

- Student database: This component stores data about each student, including their personal information, enrollment status, and academic record.
- Course catalog: This component stores information about the courses offered by the school, including the course description and availability.
- Enrollment system: This component allows students to search for and enroll in courses and updates the student database with the enrollment information.

- **Grading system:** This component allows instructors to enter grades for students and calculates students' overall grades and transcripts.
- **User authentication and authorization:** This component manages user accounts and ensures that only authorized users can access the system.

These components would likely communicate with each other through APIs or other interfaces. The specific details of the interfaces and the way that the components interact would be determined as part of the implementation design process.

Use Case Diagram:

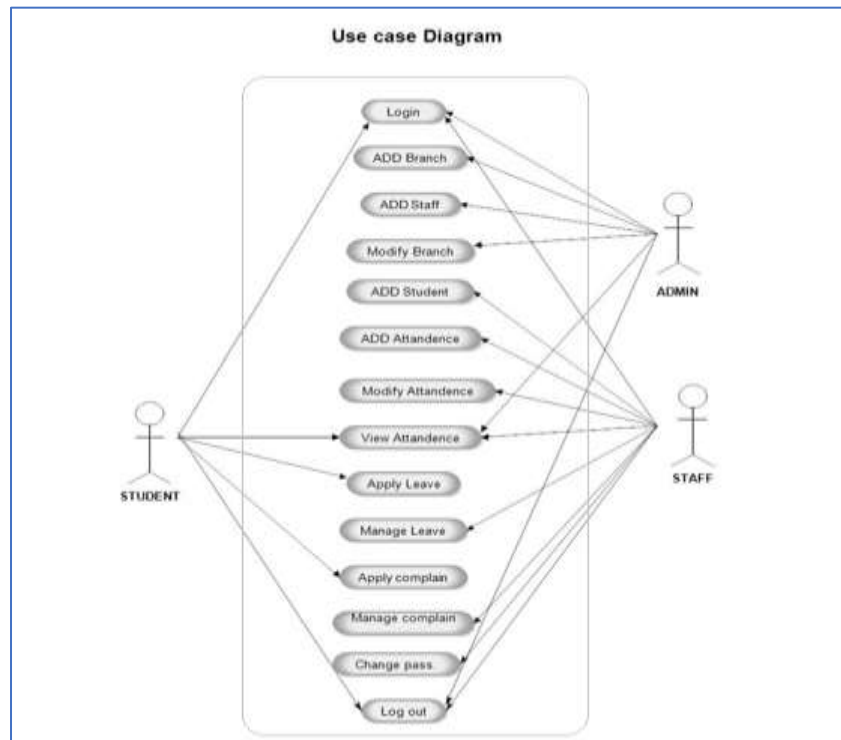


Figure 1: Use case diagram for YUVI Academy

Database design:

This type of design defines the data structures that will be used to store and organize information within the software system. Our focus is on database design because it is our database project. A database design for an Academy management system includes the following tables:

- **Students:** This table stores information about each student, including their personal details (such as name, address, and contact information) and enrollment information (such as their enrollment status and major).
- **Staff:** This table stores information about each student, including their personal details (such as name, address, and contact information)

- **Classes:** This table stores information about each class offered by the academy, including the class name, description, and section name.
- **Results:** This table stores information about which students are enrolled in which courses, as well as the grades that the students have received for those courses.
- **Users:** This table stores information about the user accounts for the system, including the username, password, and role (such as student or instructor).
- **Roles:** This table stores information about the different roles that users can have within the system (such as student, instructor, or administrator).
- **Subjects:** This table stores information about the different subjects offering in a particular class

These tables would be related to each other through foreign keys. For example, the Students table might have a foreign key to the Roles table to indicate each student's role within the system. The specific structure of the tables and the relationships between them would depend on the specific requirements of the software.

ERD Diagram:

We have total 14 tables which are fully Normalized used for our Project. There fully Entity Relationship Diagram is shown below:

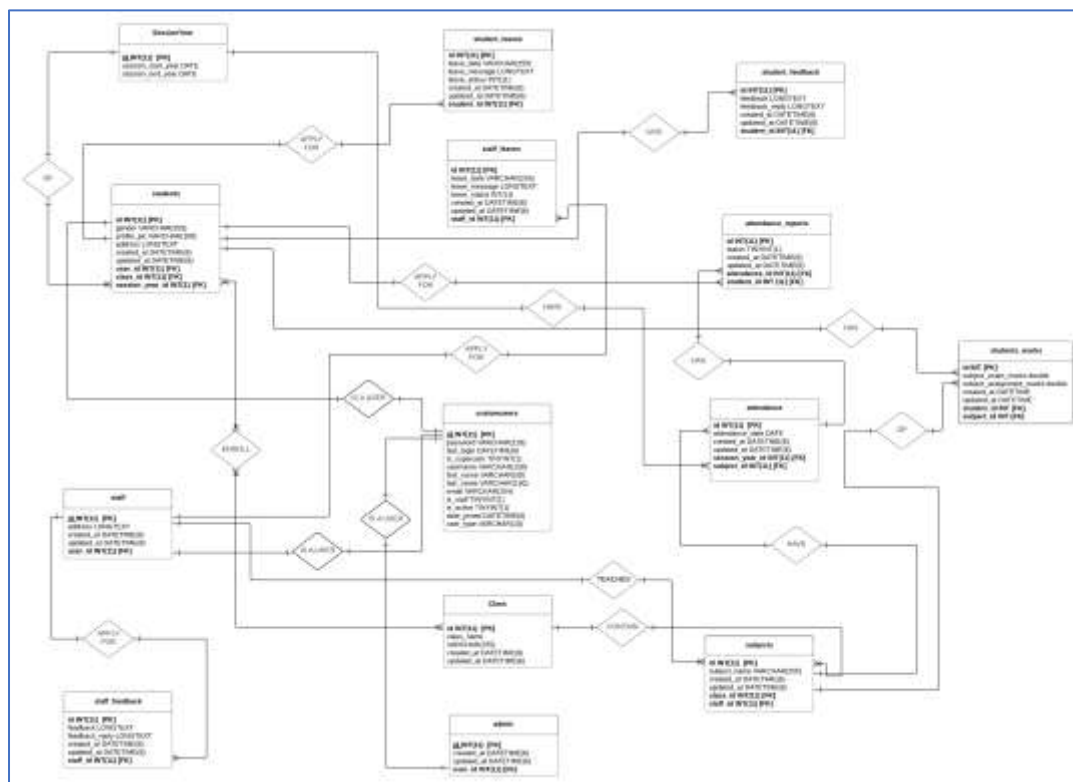


Figure 2: ERD for YUVI Academy

User interface design:

This type of design defines how the user will interact with the software, including the layout of screens and menus, and the way that users input and retrieve data.

With the goal of making UI easy and efficient for users to perform their tasks, we use latest technologies for it which includes:

- ➔ HTML
- ➔ CSS
- ➔ Bootstrap
- ➔ JavaScript Chart class

Now we will show some important screen shots of YUVI Academy software.

1. The glimpse of Login page:

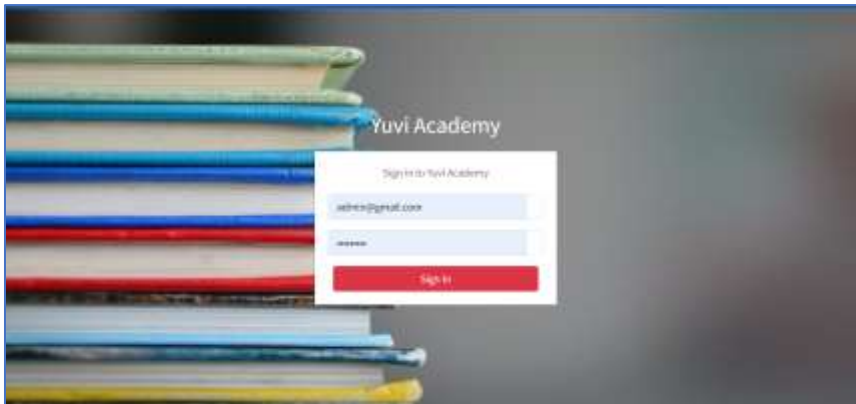


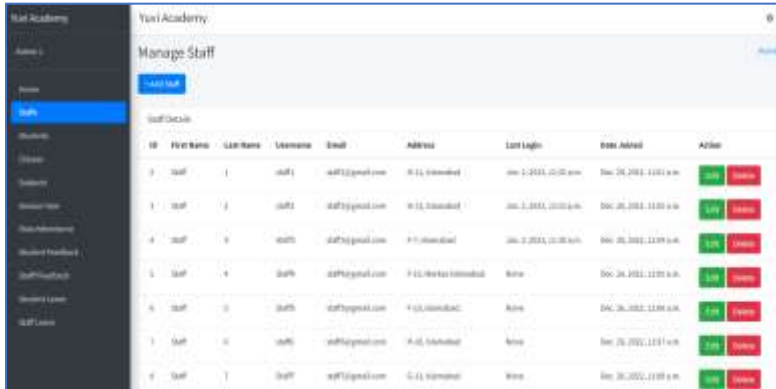
Figure 3: login page

2. Admin dashboard:



Figure 4: Admin dashboard

3. Manage Staffs



Manage Staff

Staff Details

ID	First Name	Last Name	Username	Email	Address	Last Login	Date Added	Action
1	Staff	1	staff1	staff1@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
2	Staff	2	staff2	staff2@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
3	Staff	3	staff3	staff3@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
4	Staff	4	staff4	staff4@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
5	Staff	5	staff5	staff5@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
6	Staff	6	staff6	staff6@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
7	Staff	7	staff7	staff7@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
8	Staff	8	staff8	staff8@gmail.com	111, Islamabad	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete

Figure 5: manage staff

4. Manage students:

You Academy

Manage Student

Home

Students

Subjects

Teachers

Reports

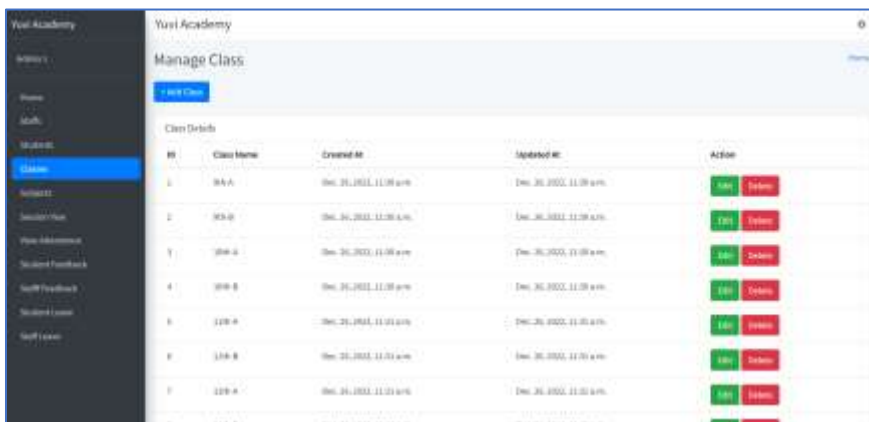
Settings

Student Details

ID	Username	Email	Address	Gender	Start Year	End Year	Course	Last Login	Date Added	Action
student1	student1@gmail.com	111, Islamabad	Male	Aug 1, 2022	April 15, 2023	100-A	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student2	student2@gmail.com	111, Islamabad	Male	Aug 1, 2022	April 15, 2023	100-A	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student3	student3@gmail.com	111, Islamabad	Female	Aug 1, 2022	April 15, 2023	100-B	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student4	student4@gmail.com	111, Islamabad	Female	Aug 1, 2022	April 15, 2023	100-A	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student5	student5@gmail.com	111, Islamabad	Male	Aug 1, 2022	April 15, 2023	100-B	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student6	student6@gmail.com	111, Islamabad	Male	Aug 1, 2022	April 15, 2023	100-A	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student7	student7@gmail.com	111, Islamabad	Male	Aug 1, 2022	April 15, 2023	100-B	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	
student8	student8@gmail.com	111, Islamabad	Male	Aug 1, 2022	April 15, 2023	100-A	Dec 1, 2022, 10:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete	

Figure 6: Manage students

5. Manage classes:



Manage Class

Class Details

ID	Class Name	Created At	Updated At	Action
1	100-A	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
2	100-B	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
3	100-A	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
4	100-B	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
5	100-A	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
6	100-B	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
7	100-A	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete
8	100-B	Dec 26, 2022, 11:00 a.m.	Dec 26, 2022, 11:00 a.m.	Edit Delete

Figure 7: Manage Classes

6. Manage Subjects:

ID	Subject Name	Class	Staff	Created At	Updated At	Action
1	Physics	9th-A	Staff 1	Dec. 15, 2022, 12:13 p.m.	Dec. 20, 2022, 12:18 p.m.	Edit Delete
2	Chemistry	9th-A	Staff 2	Dec. 15, 2022, 7:00 p.m.	Dec. 17, 2022, 12:40 p.m.	Edit Delete
3	Chemistry	10th-A	Staff 4	Dec. 15, 2022, 7:00 p.m.	Jan. 2, 2023, 11:13 a.m.	Edit Delete
5	Chemistry	9th-B	Staff 1	Dec. 16, 2022, 7:00 p.m.	Jan. 2, 2023, 11:13 a.m.	Edit Delete
12	Chemistry	10th-A	Staff 2	Dec. 16, 2022, 7:17 p.m.	Jan. 2, 2023, 11:13 a.m.	Edit Delete
13	Physics	10th-A	Staff 1	Dec. 16, 2022, 7:40 p.m.	Jan. 2, 2023, 11:17 a.m.	Edit Delete
14	Physics	10th-B	Staff 14	Dec. 17, 2022, 12:42 p.m.	Jan. 2, 2023, 11:17 a.m.	Edit Delete

Figure 8: Manage Subjects

7. View Attendance:

View Attendance

Subject: Physics

Session Year: Aug. 1, 2022 to April 15, 2023

Select Attendance Date

Attendance Date: 2023-01-02

Report Student Data

Student Attendance:

Student 1 Present 1

Copyright © 2022, Yust Academy. All rights reserved.

Figure 9: View Attendance

8. Feedbacks:

ID	Student ID	Student Name	Student Session	Message	Sendes On	Reply
1	18	Student 2	Aug. 1, 2022 - April 15, 2023	the school is nice	Dec. 26, 2022, 5:52 p.m.	Thank you dear
2	18	Student 2	Aug. 1, 2022 - April 15, 2023	no	Dec. 26, 2022, 6:00 p.m.	Any you want to be a
3	17	Student 1	Aug. 1, 2022 - April 15, 2023	best school	Jan. 2, 2023, 12:28 p.m.	ok
4	17	Student 1	Aug. 1, 2022 - April 15, 2023	best	Jan. 2, 2023, 12:38 p.m.	ok
5	17	Student 1	Aug. 1, 2022 - April 15, 2023	best	Jan. 2, 2023, 12:38 p.m.	Reply

Figure 10: Feedbacks of students and staff

9. Staff Dashboard:



Figure 11; staff dashboard

10. Take attendance:

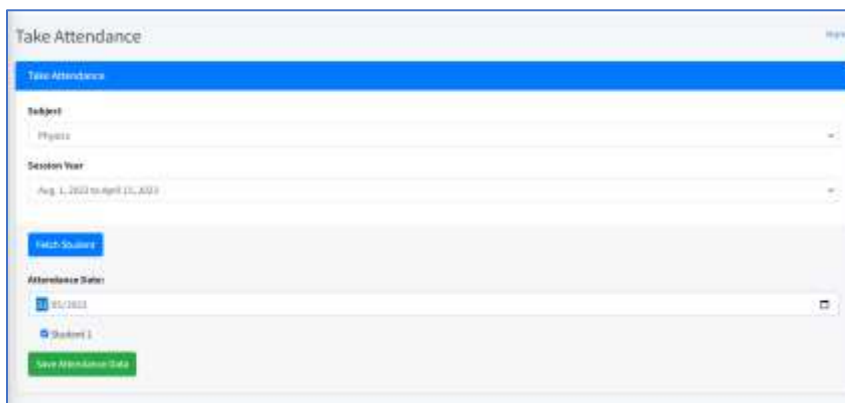


Figure 12: staff takes attendance

11. Enter Results:

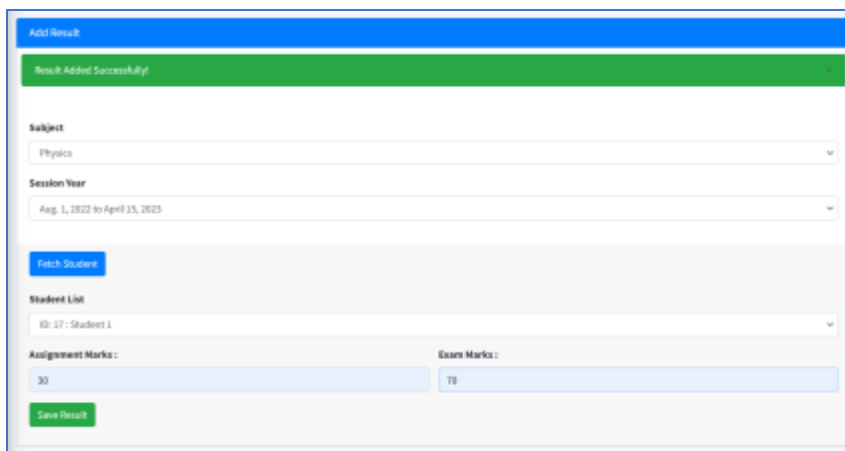


Figure 13: staff add result

12.Students Dashboard:



Figure 14: student dashboard

13.View attendance:

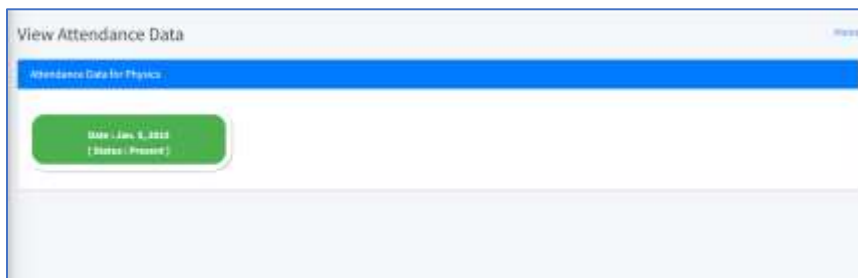


Figure 15: student view attendance

14.View results:

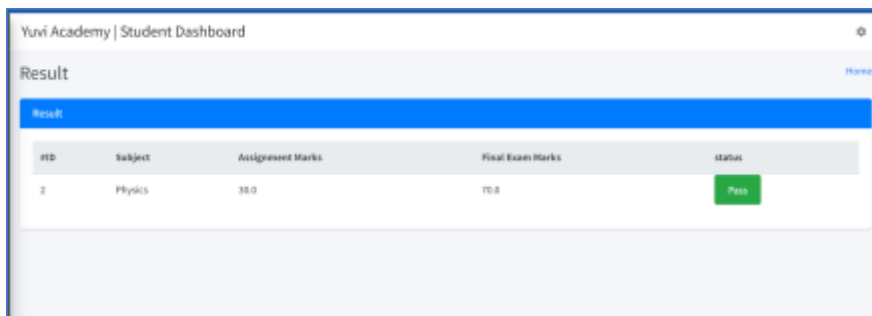


Figure 16: student view result

- Here we set a condition of at least 45% combined marks of assignment and final exam to pass.

4.Implementation design:

This type of design defines the specific details of how the software will be implemented, including the choice of programming languages, libraries, and frameworks.

Here are some general technologies we use follow to implement a academy management system:

- ➔ **Django web framework:** Django is a free and open-source web framework written in Python. Django follows the "batteries included" philosophy and provides many built-in features such as authentication, database ORM, and a templating system.
- ➔ **MySQL Database:** MySQL is a free and open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) to manage data. It is a popular choice for web applications and is widely used in the industry.
- ➔ **VS Code:** Visual Studio Code (VS Code) is a free, open-source code editor. It is best editor
- ➔ **GitHub:** GitHub is a web-based platform for version control and collaboration on software projects. It is based on the Git version control system, which allows developers to track changes to their code and collaborate with others on a project.

Testing:

It is an important part of the software development process, as it helps to identify defects, bugs, and other issues that may impact the functionality or performance of the software. We tested our project by looking at the MySQL workbench whether our data is storing in our data base or not.

Now we will show the data base entries to test whether the data was entered in the database or not when we performed the above tasks in UI section.

1. When staff mark the attendance

	id	status	created_at	updated_at	attendance_id_id	student_id_id
	8	1	2023-01-02 12:25:38.364157	2023-01-02 12:25:38.364157	4	5
	9	1	2023-01-02 12:25:38.380236	2023-01-02 12:25:38.380236	4	9
	10	1	2023-01-02 12:27:07.197354	2023-01-02 12:27:07.197354	5	5
	11	0	2023-01-02 12:27:07.205346	2023-01-02 12:27:07.205346	5	9
	12	1	2023-01-02 12:27:07.253313	2023-01-02 12:27:07.253313	6	5
	13	0	2023-01-02 12:27:07.261308	2023-01-02 12:27:07.261308	6	9
	14	1	2023-01-03 07:25:27.164436	2023-01-03 07:25:27.164436	8	1
	NULL	NULL	NULL	NULL	NULL	NULL

Figure 17: Attendance table

2. When staff add the result

	id	subject_exam_marks	subject_assignment_marks	created_at	updated_at	student_id_id	subject_id_id
	1	85	8.5	2022-12-27 18:00:49.070445	2022-12-27 18:00:49.070445	10	27
	2	70	30	2023-01-03 07:26:24.548051	2023-01-03 07:26:24.548051	1	1
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 18: Result table

3. When staff /students give feedback

Result Grid						
Filter Rows:		Edit:		Export/Import:		Wrap Cell Contents: <input type="checkbox"/>
	id	feedback	feedback_reply	created_at	updated_at	student_id_id
▶	1	the school is nice	thank you dear	2022-12-26 17:58:53.999525	2022-12-26 17:58:53.999525	2
	2	hi	luv you mustafa bhai	2022-12-26 17:58:53.999525	2023-01-03 05:47:55.733683	2
	3	best school	ok	2023-01-02 12:29:31.713235	2023-01-02 12:30:18.908928	1
	4	1st	ok	2023-01-02 12:30:55.522103	2023-01-02 12:31:40.326205	1
	5	2nd		2023-01-02 12:30:59.552317	2023-01-02 12:30:59.552317	1
✱	NULL	NULL	NULL	NULL	NULL	NULL

Figure 19: feedback table of student

4. When admin add the new student

	id	gender	address	created_at	updated_at	admin_id	course_id_id	session_year_id_id	profile_pic
	14	Female	ISB	2022-12-26 13:58:42.379129	2022-12-26 13:59:56.253884	30	7	2	NULL
	15	Female	LHR	2022-12-26 14:17:20.443311	2022-12-26 14:17:20.467774	37	7	2	NULL
	22	Female	ISB	2022-12-27 13:15:25.834435	2022-12-27 13:15:25.860139	46	7	2	
	23	Male	RwP	2022-12-27 13:16:18.739963	2022-12-27 13:16:18.771594	47	8	2	
	24	Male	ISB	2022-12-27 13:17:23.497457	2022-12-27 13:17:23.521588	48	8	2	
	25	Female	LHR	2022-12-27 13:20:06.753908	2022-12-27 13:20:06.777896	49	8	2	
	26	Female	ISB	2022-12-27 13:22:03.815163	2022-12-27 13:22:03.831115	50	7	2	
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 20: student table

5. When admin add new staff

	id	address	created_at	updated_at	admin_id
	10	G-8, Islamabad	2022-12-26 11:12:10.355385	2022-12-26 11:12:10.371052	11
	11	G-10, Islamabad	2022-12-26 11:13:17.329579	2022-12-26 11:13:17.345438	12
	12	H-9, Islamabad	2022-12-26 11:14:13.801255	2022-12-26 11:14:13.817591	13
	13	I-8, Markaz Islamabad	2022-12-26 11:15:08.230910	2022-12-26 11:15:08.246901	14
	14	I-10, Islamabad	2022-12-26 11:16:21.032796	2022-12-26 11:16:21.048765	15
	15	F-6, Islamabad	2022-12-26 11:17:23.038160	2022-12-26 11:17:23.061851	16
	16	isb	2022-12-26 14:09:23.239160	2022-12-26 14:09:23.262769	34
✱	NULL	NULL	NULL	NULL	NULL

Figure 21: staff table

The above-mentioned snippets are the just a few elaborations of what we did to complete testing of our project by adding, deleting, and updating the data.

Conclusion:

Through all the aspects being discussed, at the end we have successfully developed a project named Academy Management System i.e., YUVI Academy. This system will make the academies system less laborious and well managed. We can expect the positive changes in the system to be involved in this system. This may focus towards

- ❖ A Decentralized App
- ❖ Email System for the Academy
- ❖ Online Fee and Salary Management
- ❖ Addition of a Parent Portal
- ❖ Grading system