National Academy of Science and Technology

(Affiliated to Pokhara University)

Uttar Behadi-04, Kailali, Province-07

A

Project Report

On

Online Notice Board

For the partial fulfillment of requirements for the degree of Bachelor of Computer Engineering

Under Pokhara University

Submitted To

Department of Computer Engineering

National Academy of Science and Technology

Under the Supervision of

Mr. Sunil Bahadur Bist

Lecturer, Department of Computer Engineering

Submitted By

Aasha Bhatta(18070119)
Gyanu Awasthi(18070129)
Hem Raj Upadhyaya(18070131)
Sushil Kunwar(18070162)

BE Computer 8th Semester

Sep,2022

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express our heartfelt appreciation to all those who provided us the feasibility to complete this project. Special gratitude that we give to our 8th semester project supervisor Mr. Sunil Bahadur Bist, Lecturer, Department of Computer Engineering whose important contribution to simulating, suggestions and encouragement which helps us to complete our project.

We are also very grateful and extend our sincere thanks to the University, college administration, our respected Principal Sir and respected lecturers of the department of Computer Engineering of National Academy of Science and Technology (NAST) for their cooperation, guidance and support in course of project development.

Any accomplishment requires the effort of many people and this work is not different. Last but not the least, many thanks to our family, friends, team members, college staffs, librarian and to each and every people whose diligent effort and support makes this project possible. Many books, internet websites, tutorials on YouTube and Google were also the source of our project development.

Project Members

Aasha Bhatta(18070119)

Gyanu Awasthi(18070129)

Hem Raj Upadhyaya(18070131)

Sushil Kunwar(18070162)

ABSTRACT

In today's modern colleges, there should be a platform where people can communicate and can exchange information. Sometimes, there is a big gap between different departments of a college and students don't get information on time. There should be a centralized platform where students can check information and notices. Online Notice Board is a service that provides this platform. Teachers will be able to register on this application and they will get special privileges. They can post notices and information and our mobile app will sort this information by department. Students can view the information posted by teachers. There will be an admin panel to control the teacher's account. It will have the option to delete users, delete their posts, and add new users, etc.

LIST OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

NAST: National Academy of Science and Technology

RAM: Random Access Memory

ER Diagram: Entity Relationship Diagram

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheet

IDE: Integrated Development Environment

HOD: Head of Department

SQL: Structured Query Language

CHAPTER 1
INTRODUCTION

1.1 Introduction

In the modern society, many people have access to modern technology. Therefore, the wall notice board is outdated and has many limitations. Online notice board system is intended for colleges and universities where information and notices on a regular basis play a vital role in the performance. The proposed system will act as an online notice board that will make use of modern communication methodologies and techniques for information to follow. The system is planned to consist of various useful features for the said purpose. The proposed system aims to create a platform for issuing notices and sharing information among members of the college. Different users shall have different levels of access to the content.

The project was mainly motivated by our own experiences. Due to the unusual flow of notice in the college, we missed our final elective exam. This leads us in great dilemma and gives us idea of making such application that can easily make us aware about all the notices provided by the college authority.

In the context of college, there shall be four users of the software- administrator, department head, students, and teacher. The proposed system is focused for the information flow of the colleges. Here, the admin can add, remove and update the user as well as the content posted through the application. Teachers can post the notices and students get notified about the notices. The students also have the benefit of commenting or giving feedbacks.

The system is developed using java programming language is a mobile application which is engaged in providing up-to-date articles & notices and other information's for all the users or student associated with the particular college or the department. This online notice board project is very helpful for all type of users like existing users and new users. So, admin can leave and erase notification for users to read and see. With the help of free online notice board, users (teachers and students) can access the notifications and articles quickly not only in the particular premises, also wherever and whenever they need to know.

Current time is the internet time because maximum people are online for hours every day.

So, if college authority provides better online notice through this app, none of the student will miss the notices because online notice app connects the college authority, teacher and students which is very useful and seems effective also.

An online notice board is an excellent way for informing students. Online notice boards are intended for educational institutes where students have to be informed about a lot of things related to their academics. It is a system where people can leave any types of messages and notifications, for example, to advertise things, announce events or provide any information. This online notice board project is very helpful for all type of users like existing users and new users.

The main aim of this free online notice board project is to make information dissemination much easier in a paperless community as the world tends to interact with the online notice board facility as a project.

Online notice board admin can send the notification to the particular students regarding fee payments, results, any new activity happen in college campus or college fest participation, libraries dues, hostel room payments, any workshop registrations, warnings and reminders etc. For this work online notice board project is to make all work much easier and understandable to all. Online notice board usability is fully capable of passing relevant notices and announcements and keeping the users update from time to time.

The students are kept updated each time with the online notice board for college is uploaded based on their preferences with respect to the departments and categories through a notice board online. It is the applications to improve the usage of a notice board of the college by making it available online.

1.2 Problem Statement

In present scenario, students have to physically visit college notice board for getting notices. Nowadays also, many colleges are storing all of their student data in manual way. If the number of students is more, it is difficult to store all the data manually. Hence, the college authority needs help of some features so that they can maintain and store the record accurately.

In today’s world, everything is digitized and paper is being used less and less every day. How often has it happened that we miss some important notices because we have to go to

a wall and read the notices there. There are dedicated file-hosting sites and clouds used by some institutes, but there is a definite need for a dedicated notice board system. The proposed system is such a system.

1.3 Project Objectives

Every project has some objectives & goals to achieve and here are some of the objectives of our project.

- For the partial fulfilment of requirements for the degree of Bachelor in Computer Engineering.
- To design a simple user-friendly system.
- To easily keep track of the notice board by the user with respect to date and time.
- To provide accurate notice on time.
- To make information broadcasting much easier in a digital world.

1.4 Significance

- It will minimize the cost used to print papers and other supplies used for posting to a normal notice board.
- The system will reduce the energy and time the users take to spend checking the notice board.
- It will create a great interface for users to interact with posts.
- It will become the best communication method to broadcast notices and information between the college administration and students.
- This online notice board work generally intends to act as a support system for all users.

1.5 Features

- Role based module
- Push Notification
- User Friendly

1.6 Scope and Limitations

1.6.1 Scope

- It is applicable to colleges.
- Its usability is fully capable of passing relevant notices and announcements and keeping the users updated from time to time.
- The students are kept updated each time the online notice board for college is uploaded based on their preferences with respect to the departments and categories through a notice board online.
- It is one of the applications to improve the usage of the notice board of the college by making it available online.

1.6.2 Limitation

- It is only applicable to colleges.
- It requires the facility of the internet to run this app.
- The user should be registered in the college database to access the information.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

Existing System currently on our college has manual system of putting notices on the notice board. Its outdated now. No one has time to stand in a rush in order to read those notices. So, by keeping this system as reference we had decided to develop a new system for better information flow among the system through our app.

The manual wooden boards are located at key locations in various departments residing within the university. The notice boards in these departments contain vague paper based information that needs to be changed and replaced frequently because if not, this creates abundance and redundant information.

The wooden notice board is basically a wooden object, used as a source of information dissemination within the vicinity. This object serves to display information regarding competitions, seminars, test results, test announcements or any other important announcement that needs to be placed public. This method creates problems and is less efficient because of lack of information processing such as maintaining notices, timely updates and removals, addition of paper.

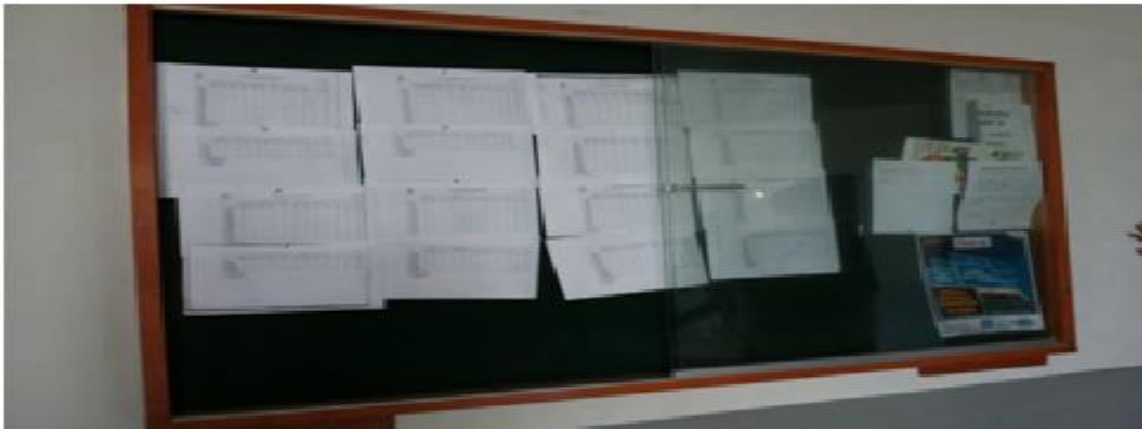


Figure 1: The manual wooden notice board

For modern and digital notice flow we had studied and taken some references from different social medias like Facebook, Google, YouTube, etc. And the developing App may somehow make ease for the student to reach the notices in time and effective manner.

CHAPTER 3
METHODOLOGY

3.1 Methodology

For our project, we have chosen the Iterative Model. In the Iterative model, the iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

The iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

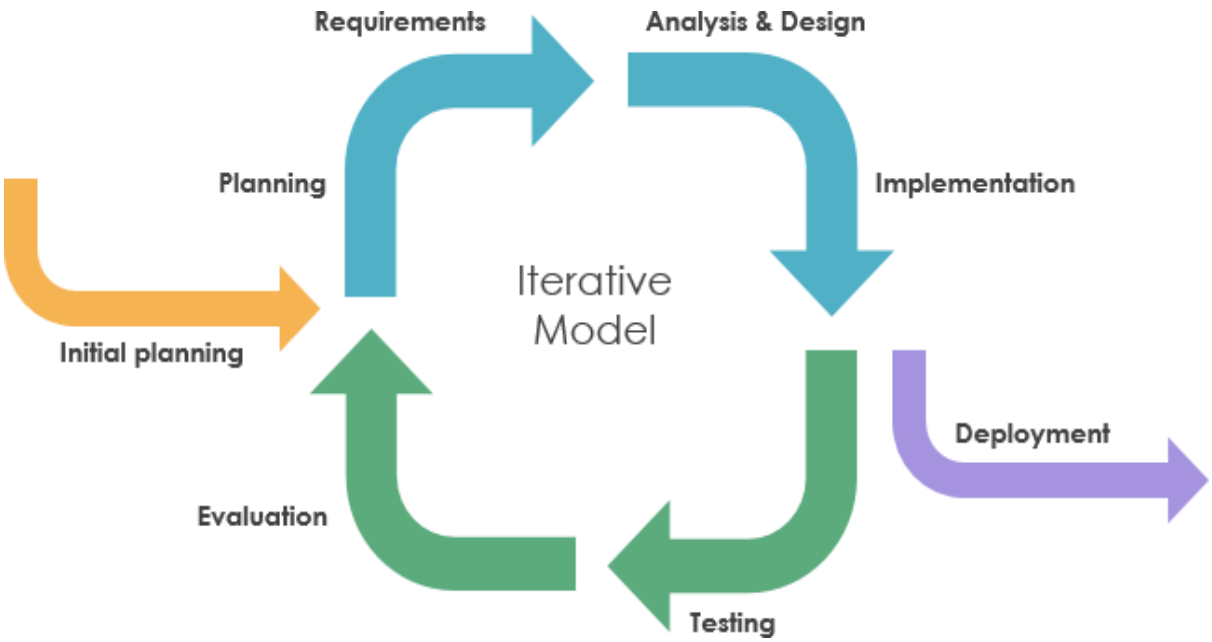


Figure 2: Iterative Model

3.2 Feasibility Analysis

A feasibility study is an analysis that takes all of a project's relevant factors into account including economics, technical, and scheduling considerations to ascertain the likelihood of completing the project successfully.

3.2.1 Technical Feasibility

Technical feasibility involves evaluation of the technical aspects of the project like hardware, software, etc. required for the proposed system. As this system is expected to be software, with multiple frame modules integrated in a single page. Some of the hardware and software requirements for the proposed system are as follows.

- **Hardware Recommendation**

  Intel i5 Core processor

  RAM 4GB and above

  64- bit OS Windows 10

- **Software Recommendation**

  Spring Boot Tool Suite

  IDE(jdk)

  Eclipse

  Postman

  Microsoft Office (for Documentation)

  Microsoft Power point

- **Tools and Techniques used**

  Front End: Java script, HTML, CSS

  API Server Side: Spring Boot

  Back end: Java, My SQL

3.2.2 Economic Feasibility

The economic costs for this project are the costs of the online domain, space, and database and registering and uploading the apps in the respective market, which are cooperatively covered by all the team members. We have determined the cost for our project would the following amount. So, the project is economically feasible.

S.N.	Elements	Price
1.	Laptop	-
2.	Pen Drive	1,500/-
3.	Internet	-
4.	RAM	-
5.	Paper	1,000/-
6.	Documentation	3,000/-
Total	Total	5,500/-

Table 1: Economic Feasibility

3.2.3 Schedule Feasibility

Our project includes the following scheduled task:

Tasks	May 19	May 29	June 15	July 25	Aug 15	Sep 20
Problem Identification						
Requirement Analysis						
System Design						
Coding & Verification						
Testing						
Implementation&Management Plan						
Maintenance &Documentation						

Table 2: Schedule Feasibility

CHAPTER 4
SYSTEM ANALYSIS AND IMPLEMENTATION

4.1 System Module Analysis

1. **Registration Module:** Users can register on the system with the help of this module.
2. **Login Module:** Users can login to the system based on their roles.
3. **Role Module:** It includes the following user role modules:
    - Admin: Admin is the person who controls, monitors and keeps the whole follow of the system. Admin can register & login/log out of the system. The main functions of admin are:
        - To keep and track down the total number of users in the system.
        - Can add, activate and deactivate any user.
        - Add new notices and send notices to the registered users.
        - Display old notices and manage them
        - Update password
        - Admin will be responsible for maintaining registered student databases.

    - HOD: Head of the department (HOD) and teacher have similar functions.
        - Registration
        - Can create categories
        - Publish notices

    - Teacher: Teacher is the information provider.
        - Can add and remove the students.
        - Should be registered first
        - Have log in and log out features.
        - Can update the profile and password.
        - Upload notices and files.

    - Student: Student is a knowledge seeker or a learner. The student has to be registered and log in into the system before he/she carries out their functions. After registering, students can log in/log out of the system. Students can view notices regarding exams, assignments, results, marks, other activities, etc.

- Student registration
- Login
- Check notices
- Update own profile and password
- Can comment and give feedback
- Log out the system

4. **Common Notice Board:** It consists of all kinds of notices. In this section, notices regarding the interest of common are published like exam notices, public holiday notices, etc. These notices are seen by every user registered in the system.

5. **Class Notice Board:** It consists of a class-wise notice board. Here, in this section, teacher publishes notices regarding the particular class. This may differ according to the program and faculty.

6. **Teacher and Student Info:** It consists of all the details of the students as well as teacher's information. The main purpose of this section is for collecting the required information about the teacher and students.

7. **Upcoming Events:** It includes all the upcoming and future events to happen in the college administration. It is used to provide notice about the future events to make everyone aware about the events and it helps conducting the events on time as scheduled.

8. **My Complaints and Feedback:** If students have any complaints against the college authority or college administration, he/she can directly post complaints as well as can give feedback using this module.

4.2 Implementation

4.2.1 Use Case Diagram of the System

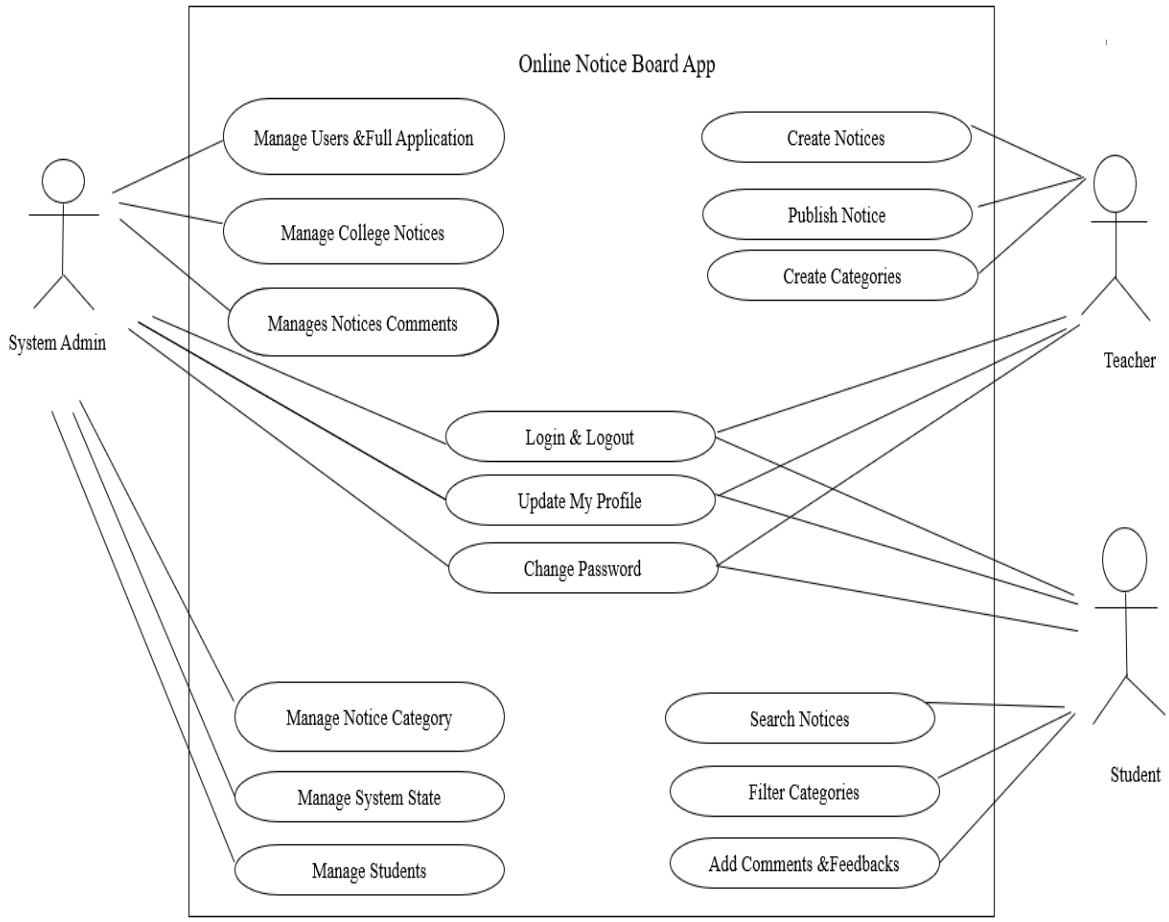


Figure 3: Use case Diagram

4.2.2 Block Diagram

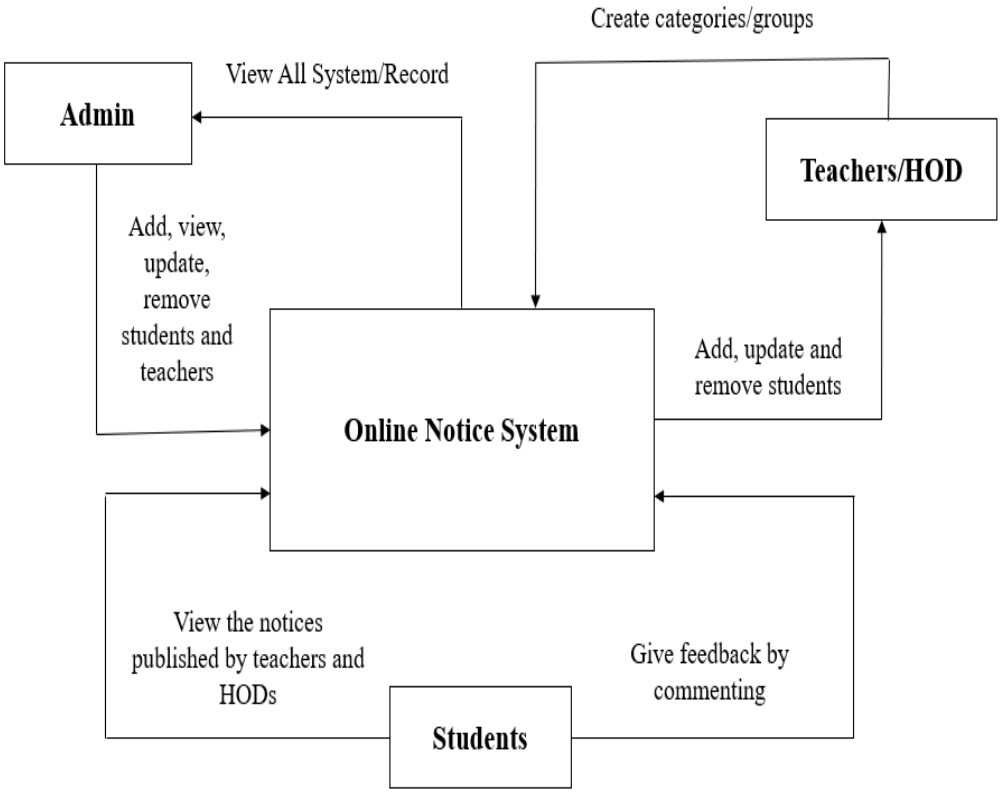


Figure 4: Block Diagram of Online Notice System

4.2.3 Flowchart

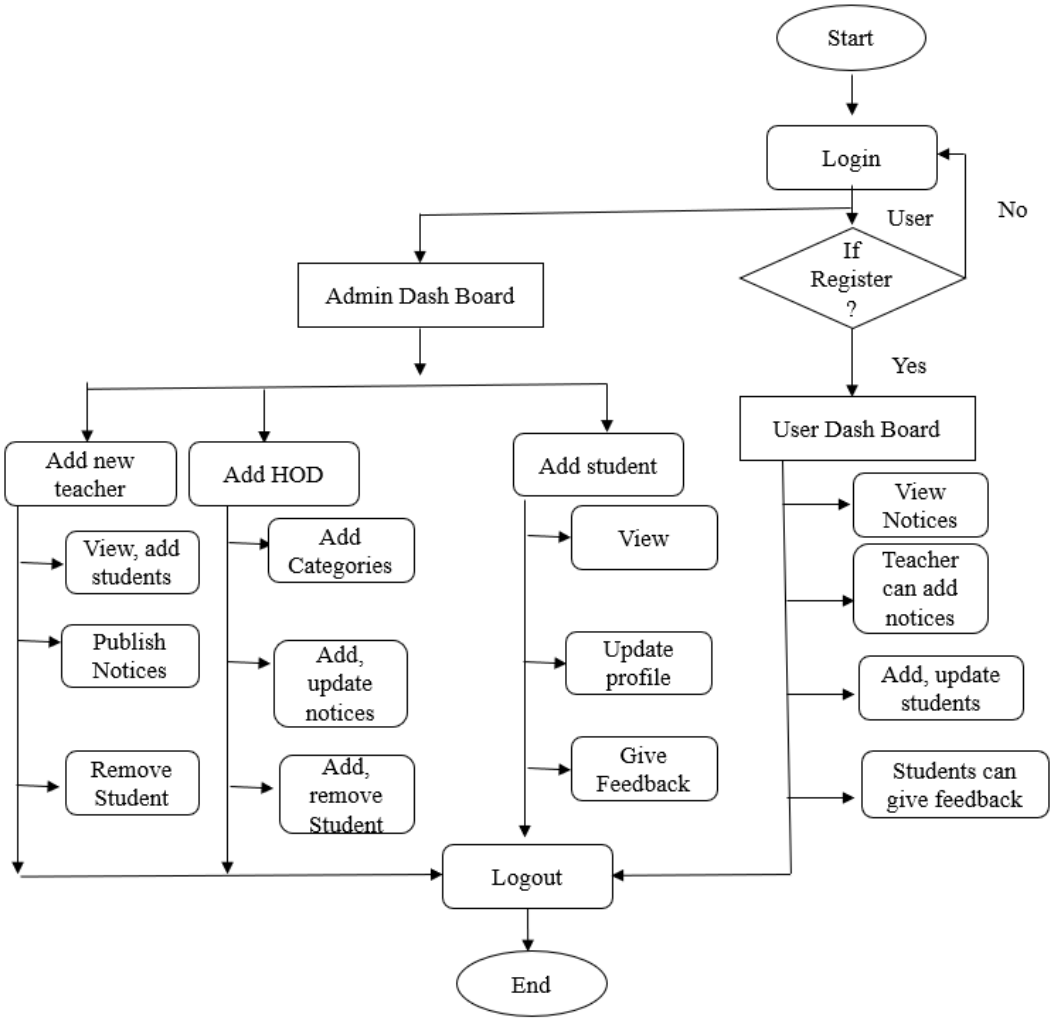


Figure 5: Flow Chart

4.2.4 ER Diagram

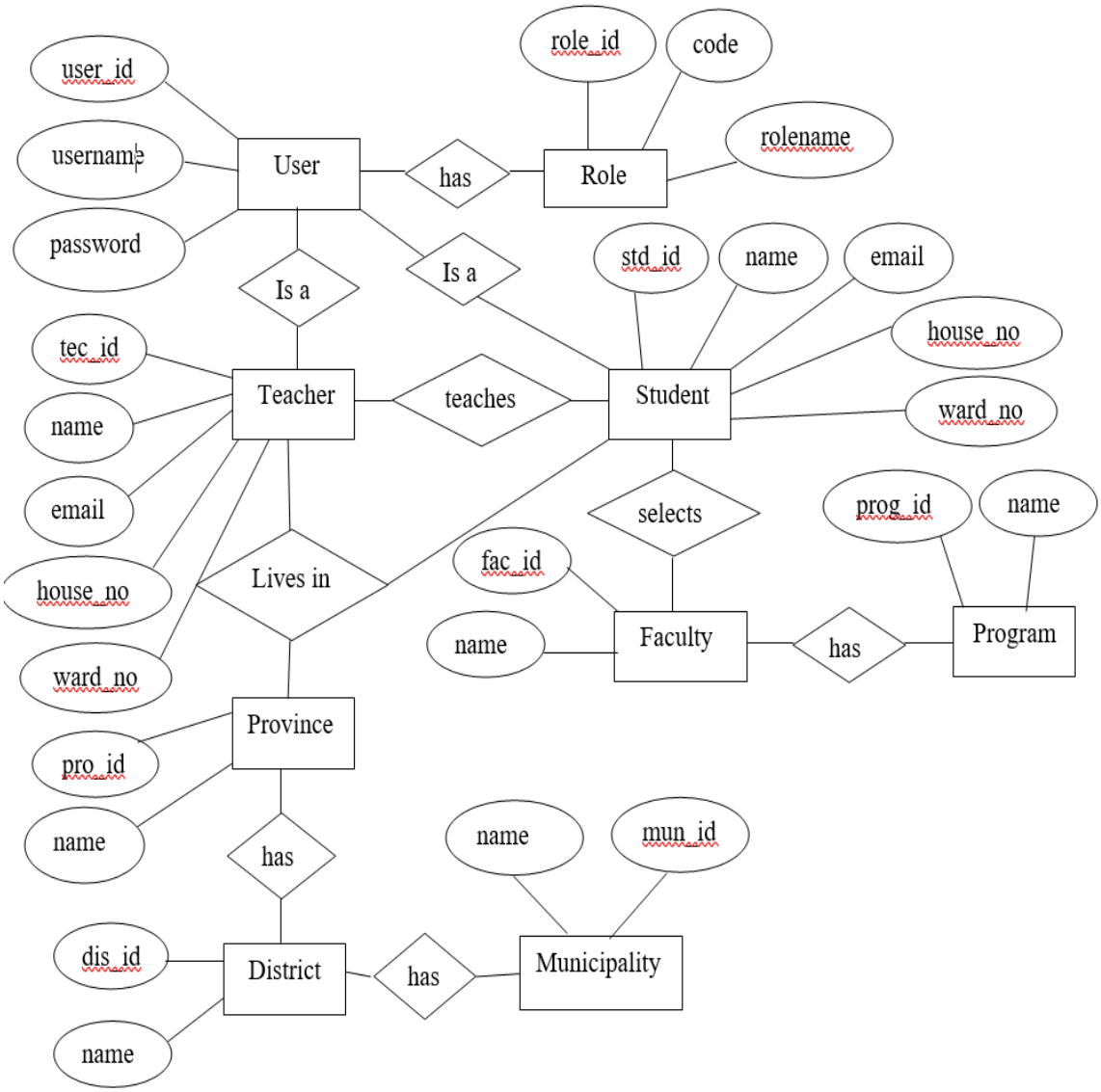
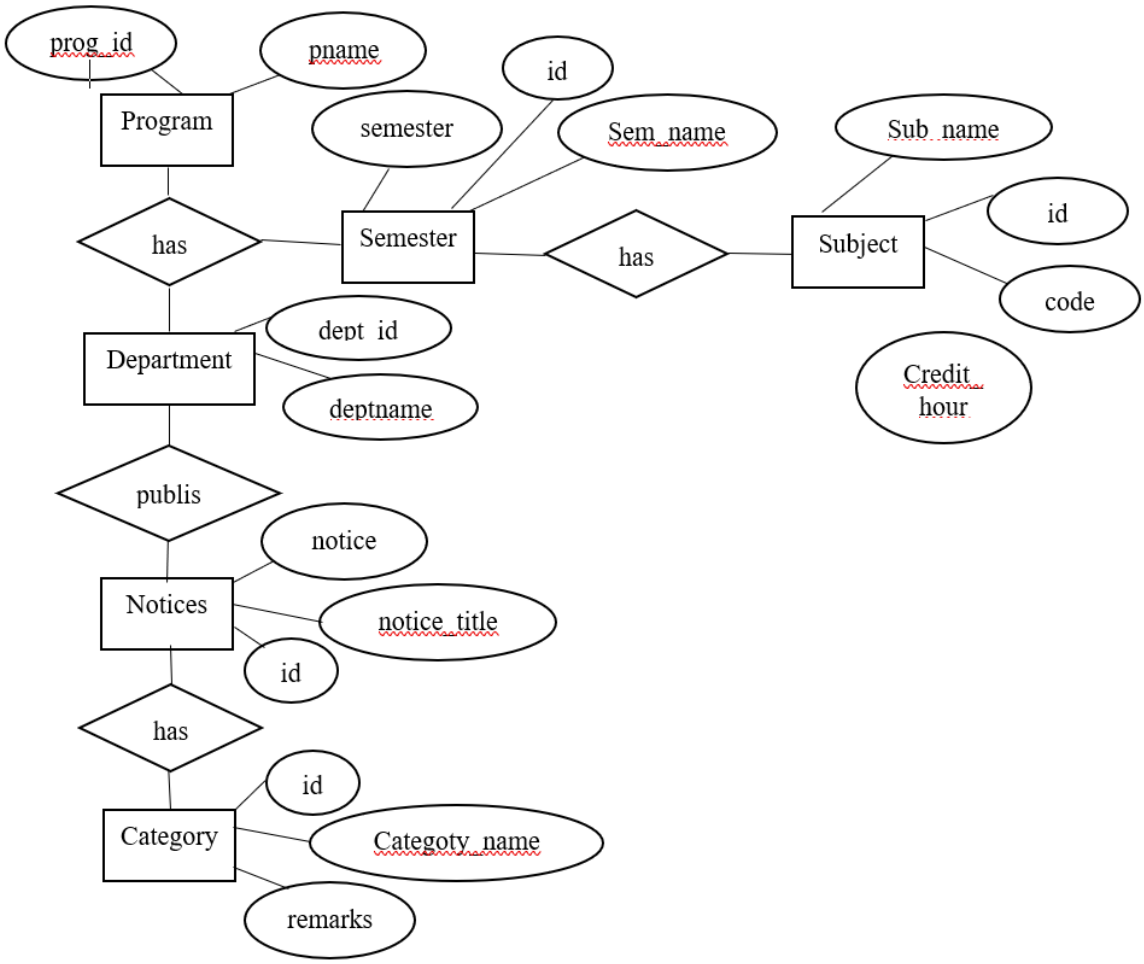


Figure 6: ER Diagram

Continued..


prog_id
pname
Program
semester
id
Sem_name
Sub_name
has
Semester
has
Subject
id
code
Credit hour
dept_id
Department
deptname
publis
notice
Notices
notice_title
id
has
id
Category
Categoty_name
remarks

4.2.5 Database Diagram

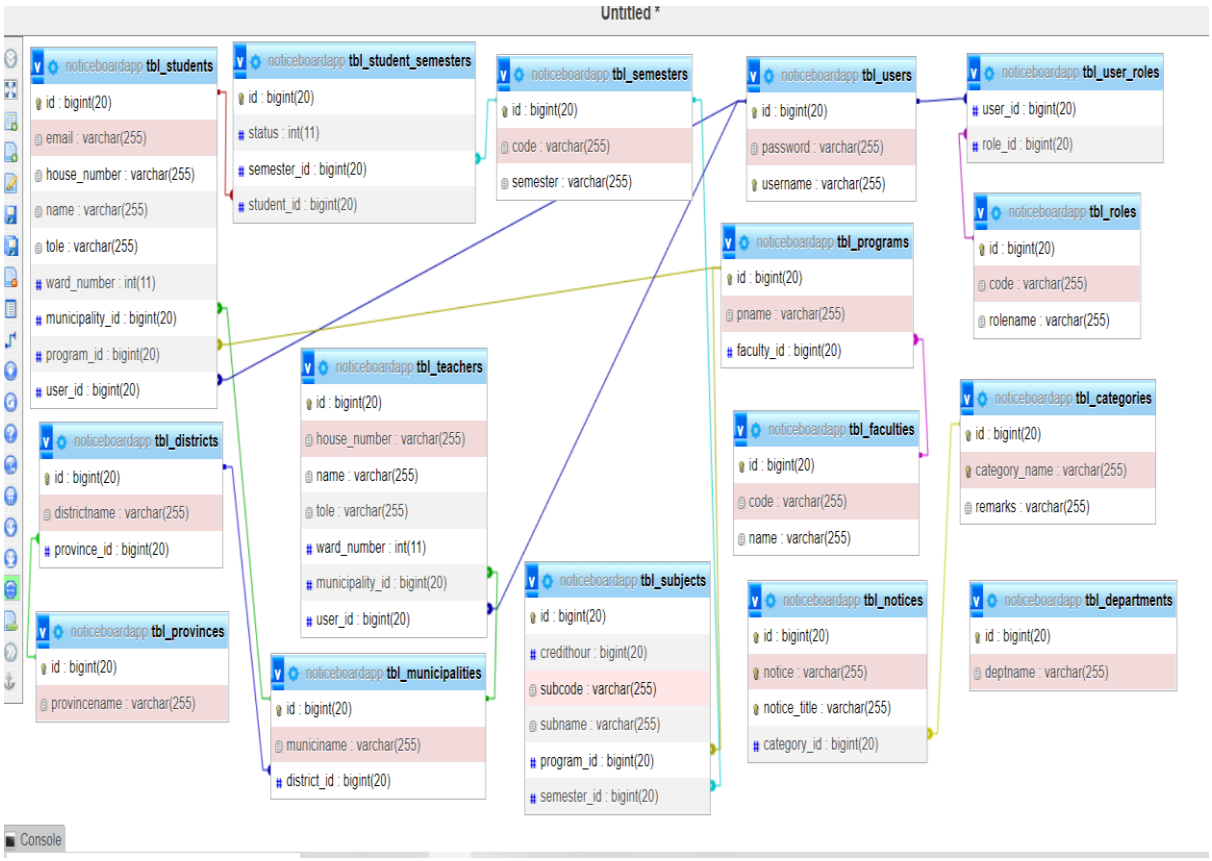


Figure 7: Database Diagram

CHAPTER 5
TESTING

Test techniques include the process of executing a program or application with the intent of finding failures and verifying that the software product is fit for use. For our project we use Black box testing and White box testing.

5.1 Black Box Testing

Black box testing is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually, test engineers are involved in black box testing.

5.2 White Box Testing

White box testing is the testing process in which tester can perform testing on the application with having internal knowledge. Usually, the developers are involved in white box testing.

- **Unit testing**

Unit testing is one of the common steps performed for every activity because it helps in removing basic and simple error.

- **Branch Testing Coverage**

It gives a wide room for testers to find quick results and helps in verifying all the possible branches in terms of lines of code. The steps offer better access to find rectify any kind of abnormal behavior in the application easily.

- **Security Testing**

It verifies any kind of unauthorized access to the system. The process helps in avoiding any kind of breach because of hacking or cracking practices.

CHAPTER 6
CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The proposed system is very effective and efficient application that serves using internet. Notices play very important role in the student life and for common people also. The proper flow of notice is required for the effective functioning of that certain program or event.

Over all we have created the application keeping the thought in mind that every student can reach to the notices easily. It means, no one has to face the lack of notice problem. This problem of notice missing can be removed using the online notice application.

The main focus of our system is to reduce the manual paper notice work by making it available online where students, teacher and college authority are connected and updated about all the notices flow of college easily using online noticeboard application.

6.2 Recommendations

- Features to record the student's attendance can be added
- Result Displaying Features can be added.

REFERENCE

[1]"Report on e-Notice App (An Android Application)", *Slideshare.net*, 2022. [Online]. Available: https://www.slideshare.net/kapoorpriyanka/report-on-enotice-app-an-android-application. [Accessed: 02- Jun- 2022].

[2]"Project Proposal on online notice board - PDFCOFFEE.COM", *pdfcoffee.com*, 2022. [Online]. Available: https://pdfcoffee.com/project-proposal-on-online-notice-board-pdf-free.html. [Accessed: 02- Jun- 2022].

[3]*Ijireeice.com*, 2022. [Online]. Available: https://ijireeice.com/wp-content/uploads/2016/04/NCAEE-50.pdf. [Accessed: 02- Jun- 2022].

[4]"Designing a college campus-based Noticeboard Android app", *Medium*, 2022. [Online]. Available: https://medium.com/@gouranshi/designing-a-college-campus-based-noticeboard-android-app-a6ef4d7ac0e9. [Accessed: 02- Jun- 2022].

[5]F. Ros, "Online Notice Board Using PHP With Source Code", *Source Code & Projects*, 2022. [Online]. Available: https://code-projects.org/online-notice-board-using-php-source-code/. [Accessed: 24- Aug- 2022]

[6]"Digital Notice Board", *SignageNepal*, 2022. [Online]. Available: https://www.signagenepal.com/digital-notice-board. [Accessed: 26- Aug- 2022]

[7]"Official Virtual Notice Board", *Sites.google.com*, 2022. [Online]. Available: https://sites.google.com/monzanet.it/official-virtual-notice-board-/home. [Accessed: 01- Sep- 2022]

[8]2022. [Online]. Available: https://m.apkpure.com/online-notice-board/com.project.onb. [Accessed: 07- Sep- 2022]

[9]"Build software better, together", *GitHub*, 2022. [Online]. Available: https://github.com/topics/noticeboard. [Accessed: 13- Sep- 2022]

[10]"10 Uses for Noticeboard App - ScreenCloud", *Screencloud.com*, 2022. [Online]. Available: https://screencloud.com/blog/10-uses-noticeboard-app. [Accessed: 18- Sep- 2022]

ANNEX I

Screenshots of User Interface

Figure: Annex 1.1 OTP Verification

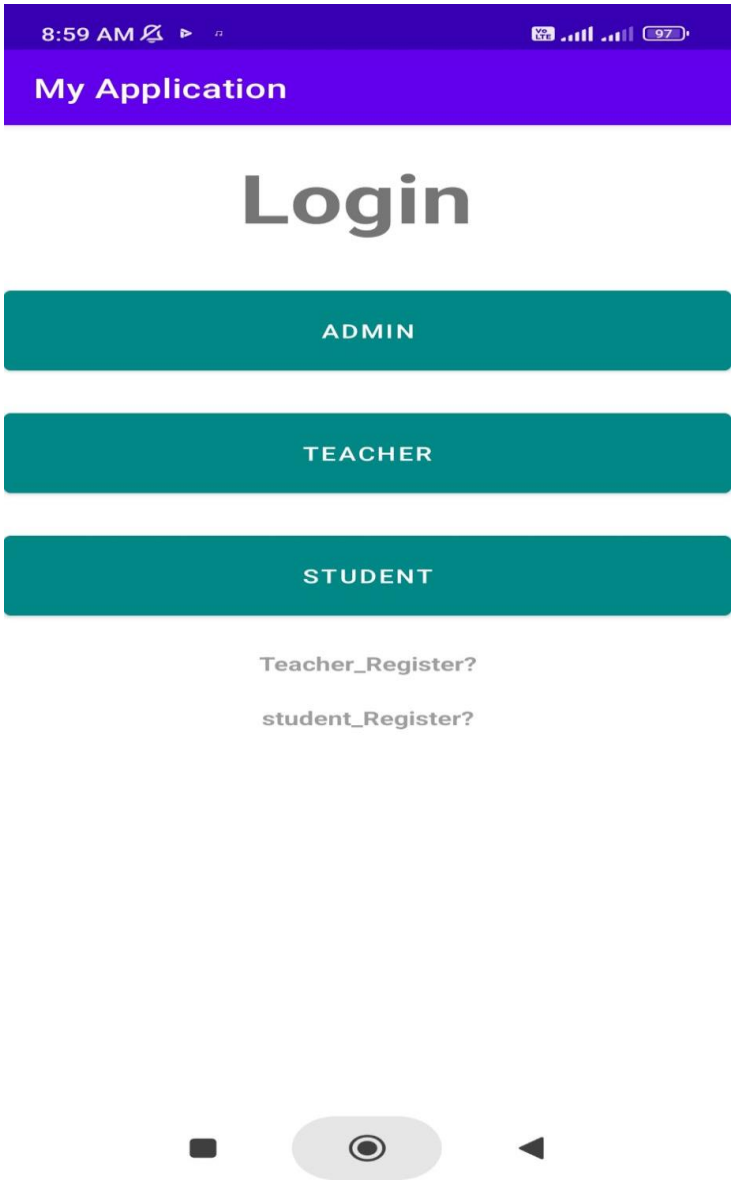


Figure: Annex 1.2 Login Page

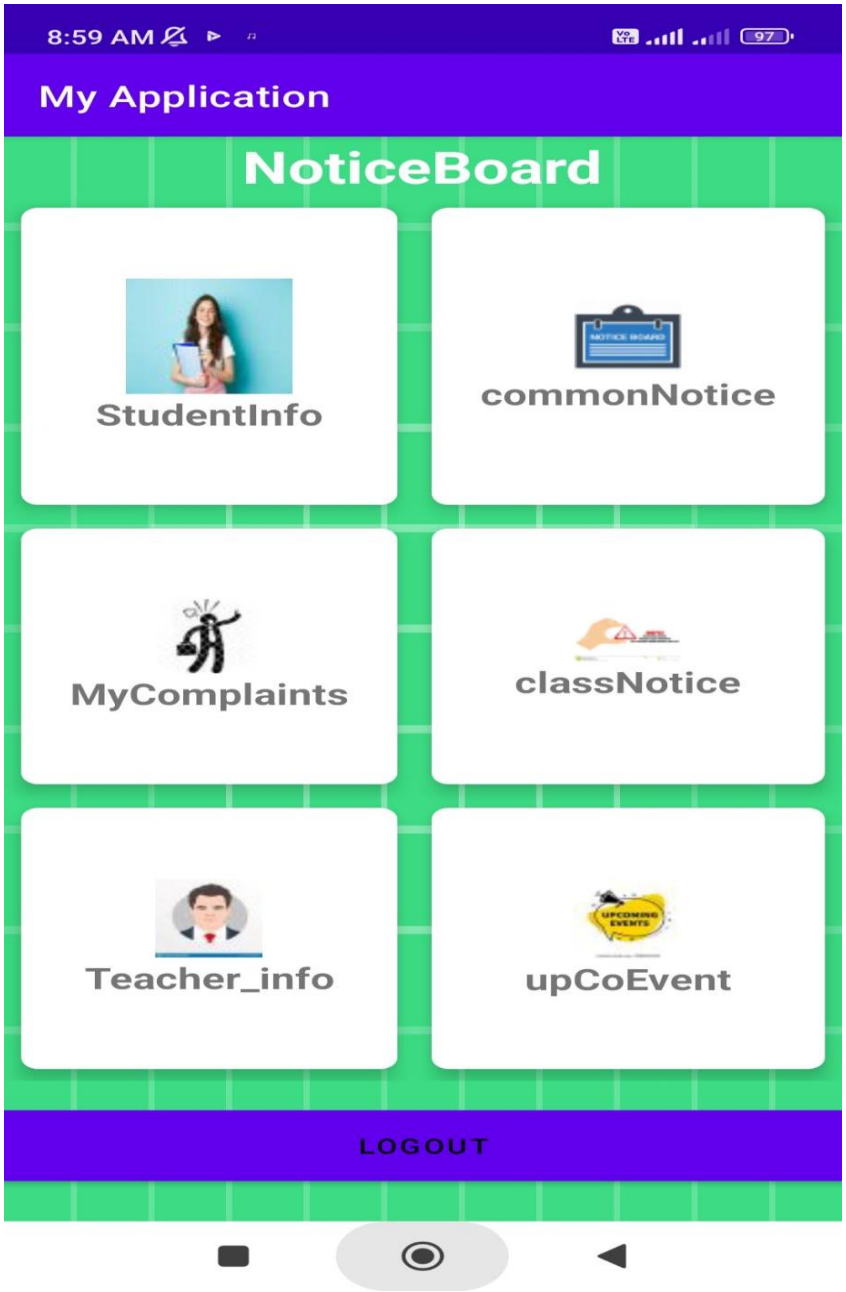


Figure: Annex 1.3 Home Page

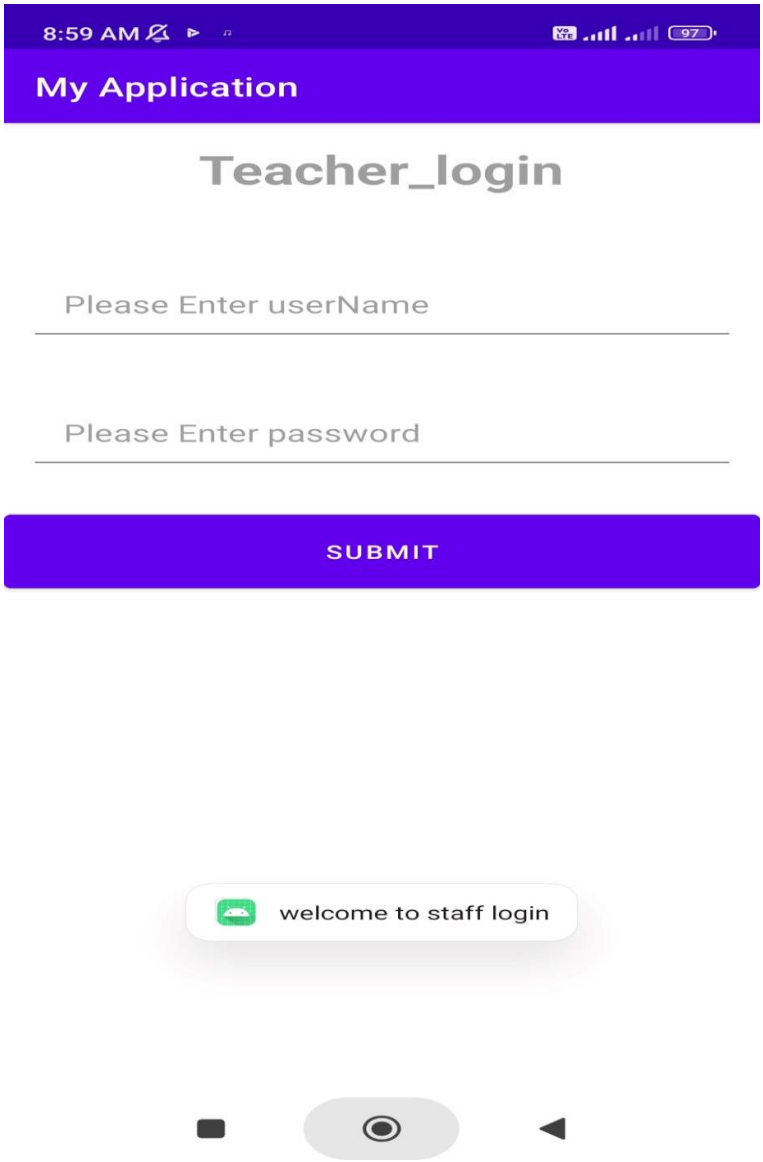


Figure: Annex 1.4 Teacher's Login Page

ANNEX II

Data in Database

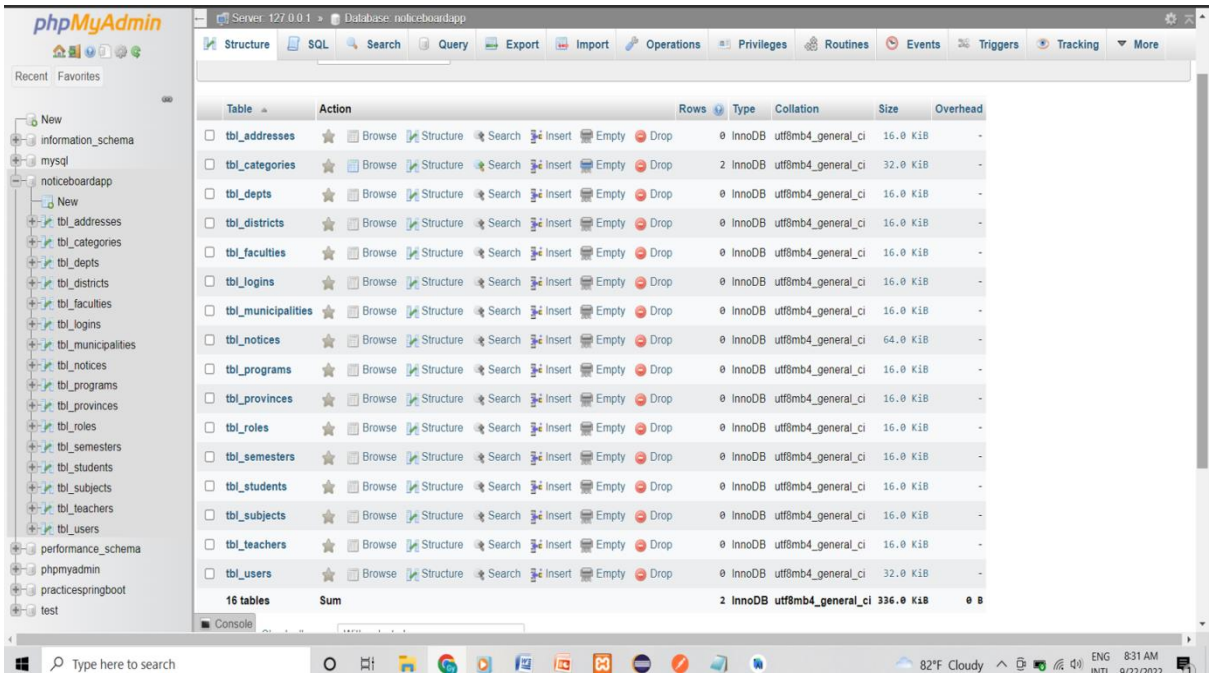


Figure: Annex 2.1Tables in Database

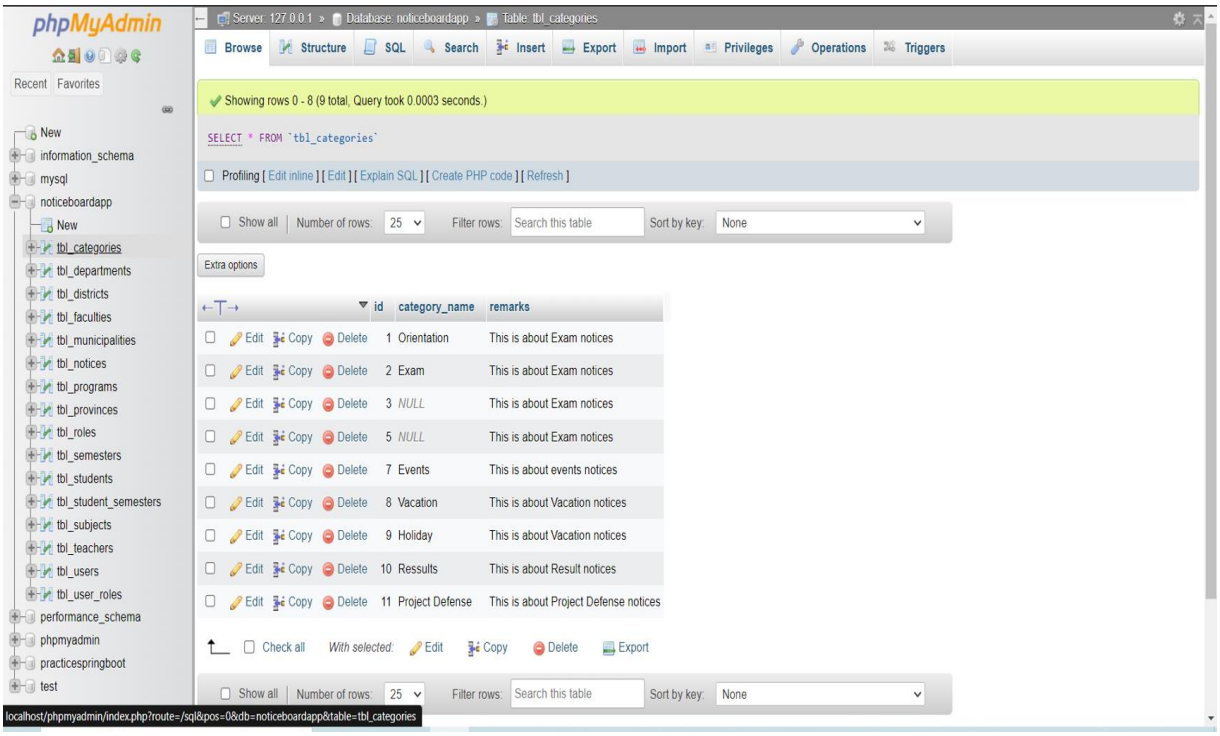


Figure: Annex 2.2 Category table

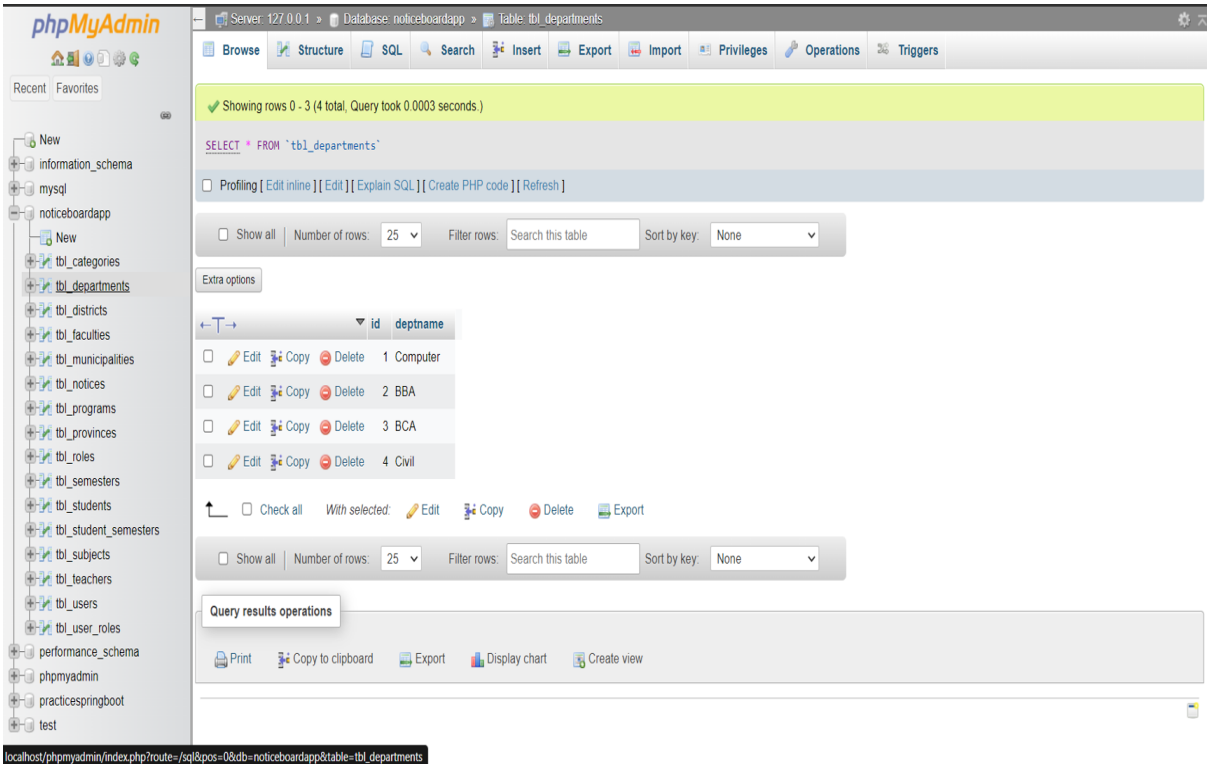


Figure: Annex 2.3 Department Table

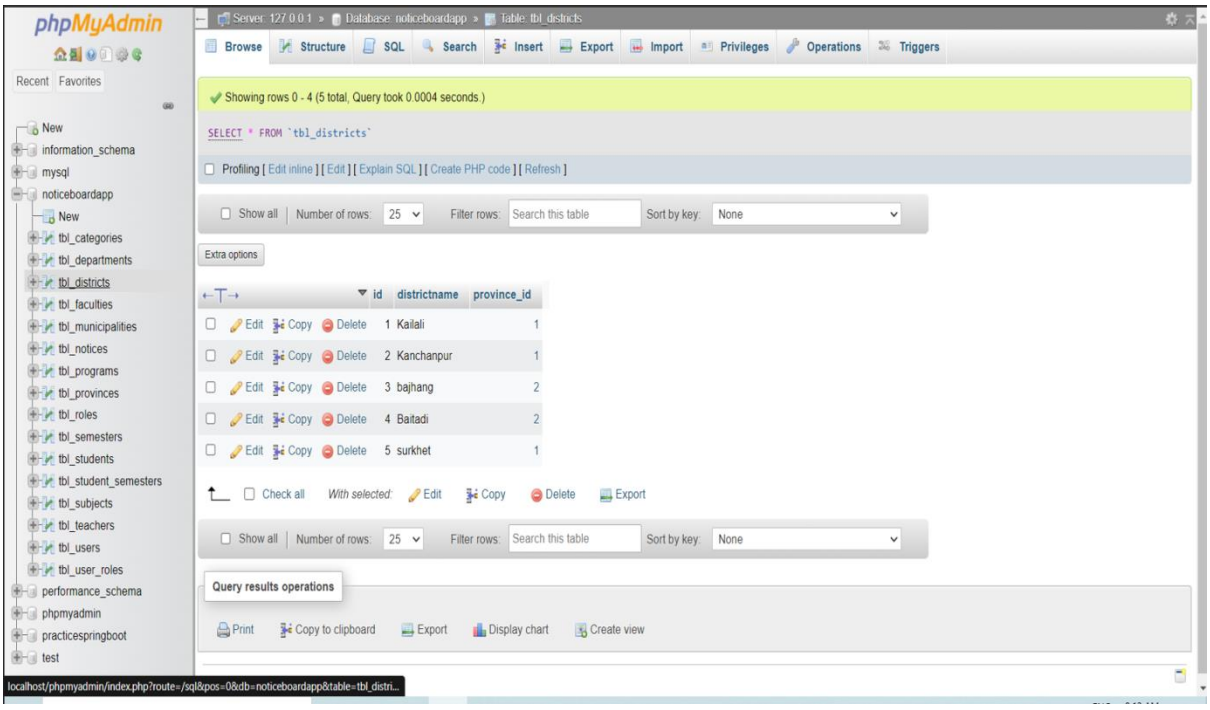


Figure: Annex 2.4 District Table

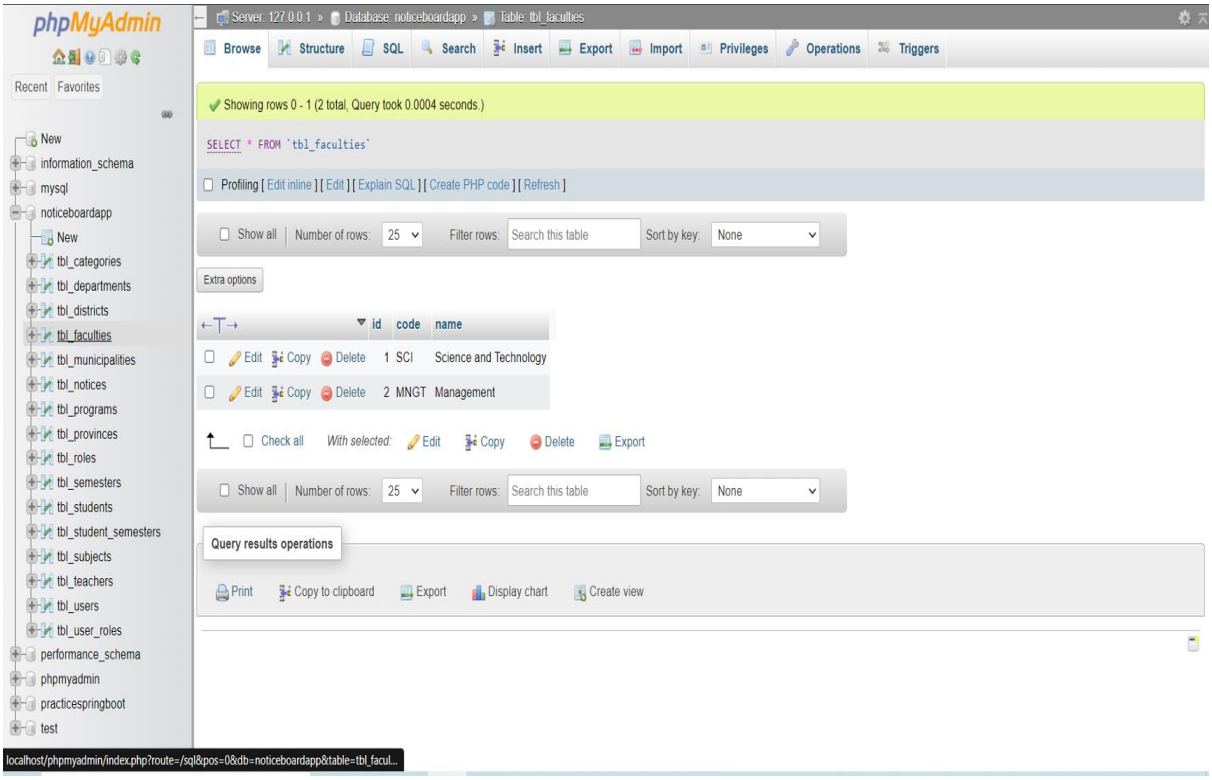


Figure: Annex2.5 Faculty Table

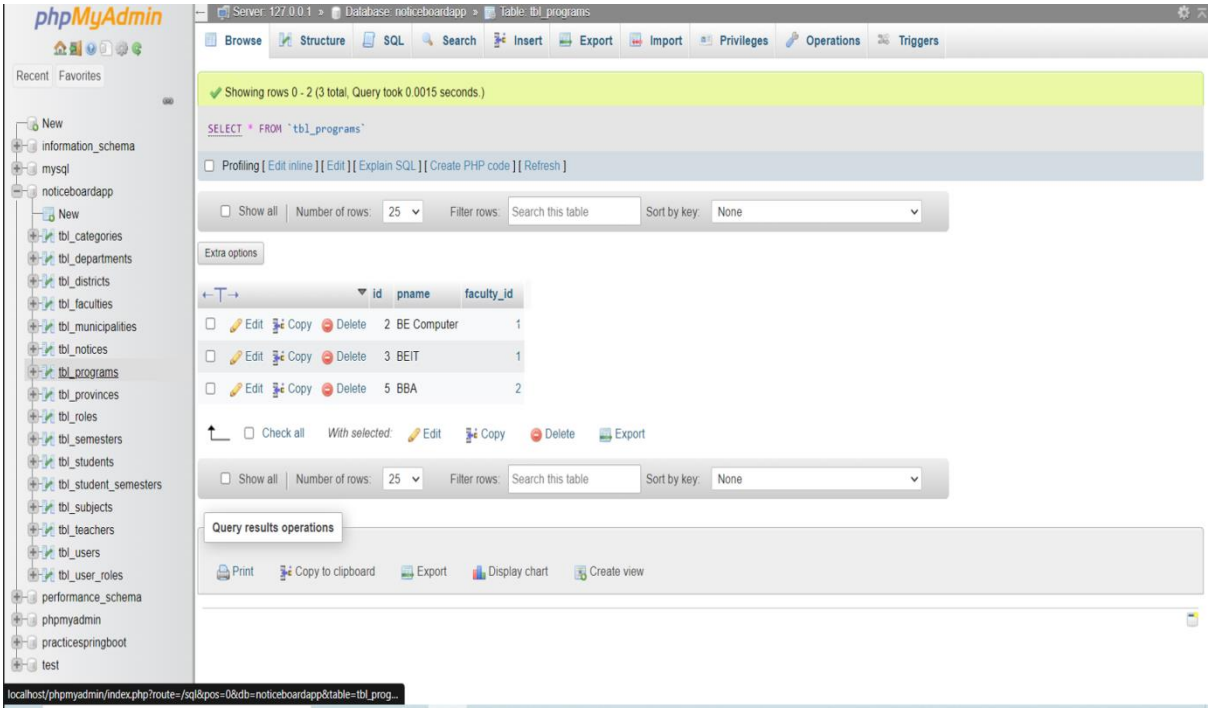


Figure: Annex 2.6 Program Table

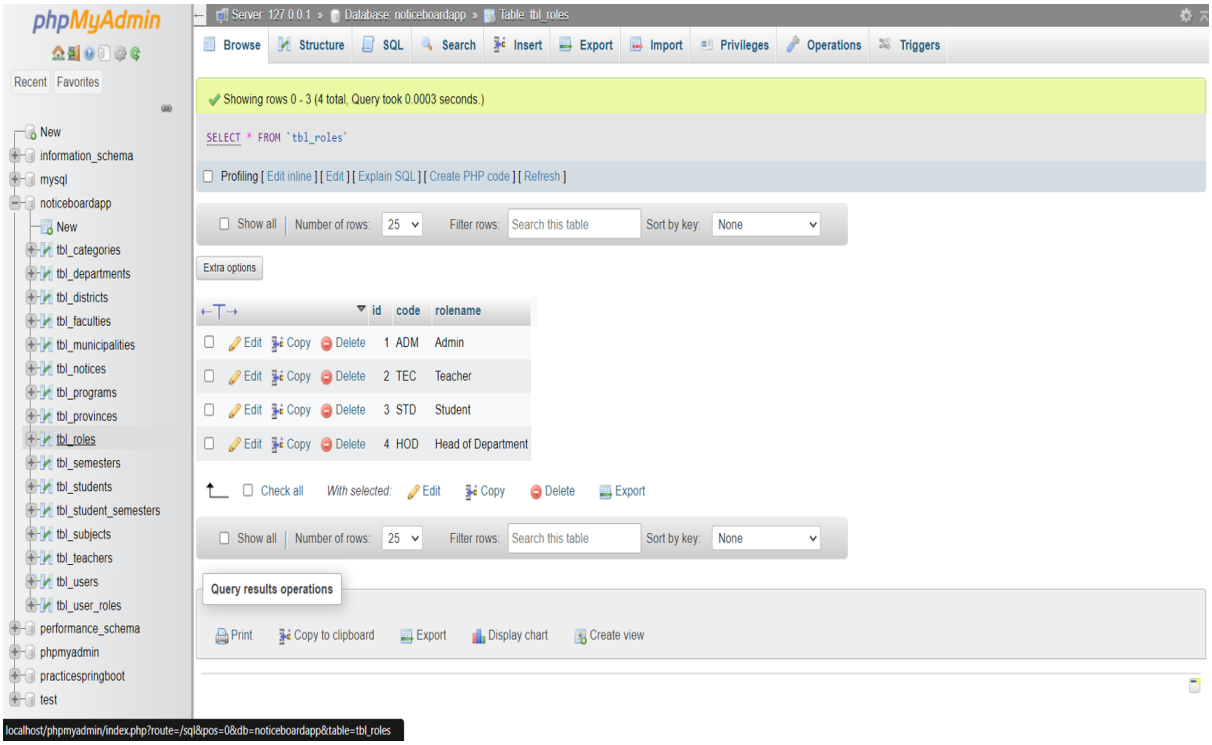


Figure: Annex2.7 Role Table

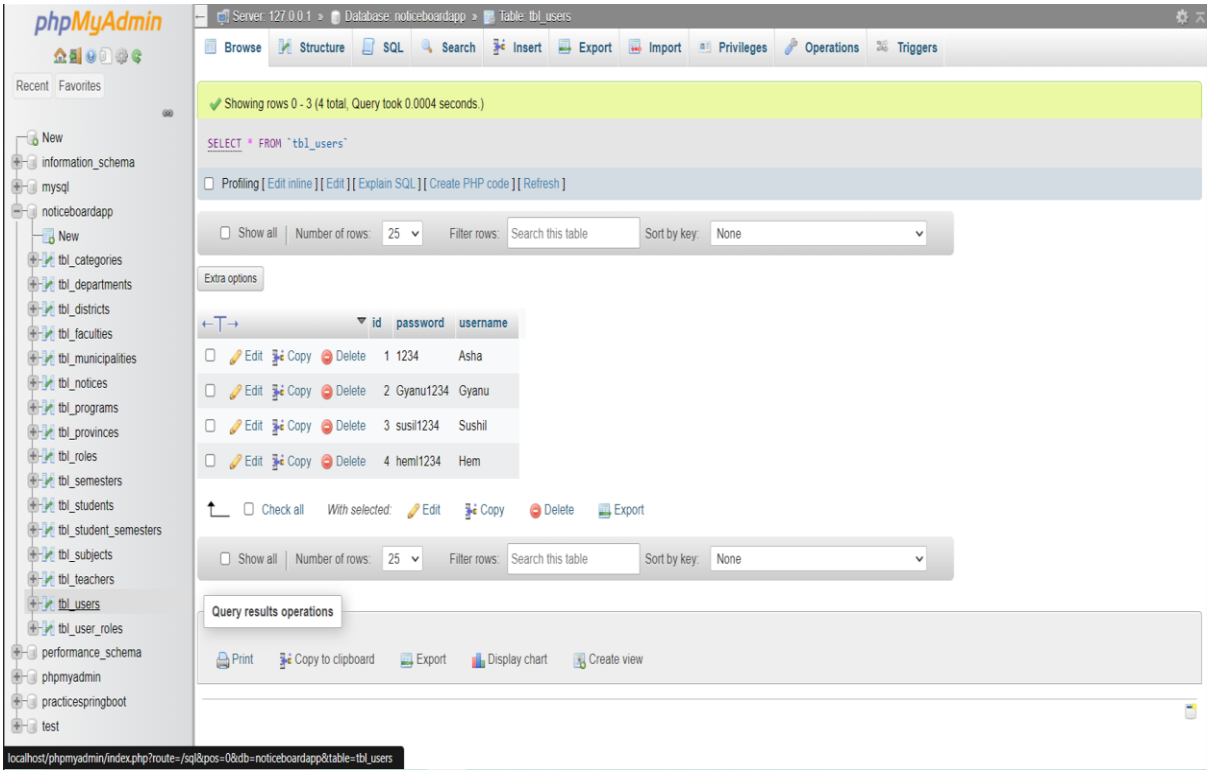


Figure: Annex2.8 User Table

ANNEX III

Backend Code

1. NoticeboardApplication.java

```
package np.com.nast.noticeboard;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class NoticeboardApplication {

	public static void main(String[] args) {
		SpringApplication.run(NoticeboardApplication.class, args);
	}

}
```

2. Role.java

```
package np.com.nast.noticeboard.entity;

import java.util.ArrayList;
import java.util.List;


@Entity
@Table(name="tbl_roles")
public class Role {
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private long id;
	private String code;
	private String rolename;

	@ManyToMany(mappedBy = "roles")
	private List<User> users = new ArrayList<>();

	public Long getId() {
		return id;
	}
	public String getCode() {
		return code;
	}
	public void setCode(String code) {
		this.code = code;
	}
```

```
	public String getRolename() {
		return rolename;
	}
	public void setRolename(String rolename) {
		this.rolename = rolename;
	}
	public void setId(long id) {
		this.id = id;
	}
	public List<User> getUsers() {
		return users;
	}
	public void setUsers(List<User> users) {
		this.users = users;
	}

}
```

3. User.java

```
package np.com.nast.noticeboard.entity;

import java.util.ArrayList;
import java.util.List;


@Entity
@Table(name="tbl_users")
public class User {
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private long id;
	@Column(unique = true)
	private String username;
	private String password;

	@ManyToMany
	@JoinTable(
			name = "tbl_user_roles",
			joinColumns = @JoinColumn(name = "user_id"),
			inverseJoinColumns = @JoinColumn(name = "role_id"))
	private List<Role> roles = new ArrayList<>();


	@OneToMany(mappedBy = "user")
	private List<Student> students = new ArrayList<>();

	@OneToMany(mappedBy = "user")
	private List<Teacher> teachers = new ArrayList<>();
```

```
	public long getId() {
		return id;
	}
	public void setId(long id) {
		this.id = id;
	}
	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public List<Role> getRoles() {
		return roles;
	}
	public void setRoles(List<Role> roles) {
		this.roles = roles;
	}
}
```

4. Notice.java

```
package np.com.nast.noticeboard.entity;


@Entity
@Table(name="tbl_notices")
public class Notice {
	@Id
	private Long id;
	@Column(unique = true)
	private String noticeTitle;
	@Column(unique = true)
	private String notice;
```

```
    @ManyToOne
    private Category category;

    public Notice() {
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getNoticeTitle() {
        return noticeTitle;
    }

    public void setNoticeTitle(String noticeTitle) {
        this.noticeTitle = noticeTitle;
    }

    public String getNotice() {
        return notice;
    }

    public void setNotice(String notice) {
        this.notice = notice;
    }

    public Category getCategory() {
        return category;
    }

    public void setCategory(Category category) {
        this.category = category;
    }


}
```

5. Category.java

```
package np.com.nast.noticeboard.entity;

import java.util.ArrayList;
import java.util.List;

import javax.persistence.Column;

@Entity
@Table(name="tbl_categories")
public class Category {
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private Long id;
	@Column(unique = true)
	private String categoryName;
	private String remarks;

	@OneToMany(mappedBy ="category")
	private List<Notice> notices= new ArrayList<>();

	public Category() {
	}

	public Long getId() {
		return id;
	}

	public void setId(Long id) {
		this.id = id;
	}

	public String getCategoryName() {
		return categoryName;
	}

	public void setCategoryName(String categoryName) {
		this.categoryName = categoryName;
	}

	public String getRemarks() {
		return remarks;
	}

	public void setRemarks(String remarks) {
		this.remarks = remarks;
	}

	public List<Notice> getNotices() {
```

```
		return notices;
	}
	public void setNotices(List<Notice> notices) {
		this.notices = notices;
	}
}
```

6. StudentController.java

```
package np.com.nast.noticeboard.controller;


@RestController
@RequestMapping("/students")
public class StudentController {
	@Autowired
	private StudentRepository studentRepo;

	//create
	@PostMapping
	public Student createStudent(@RequestBody Student student) {
		return studentRepo.save(student);
	}

	@GetMapping
	public List<Student> findAllStudent(){
		return studentRepo.findAll();
	}

	@GetMapping("/{id}")
	public Optional<Student> findStudent(@PathVariable Long id){
		return studentRepo.findById(id);
	}

	@PutMapping("/{id}")
	public Student createStudent(@RequestBody Student student,@PathVariable Long
id) {
		if(id==student.getId())
			return studentRepo.save(student);
		return null;

	}
	@DeleteMapping("/{id}")
		public void deleteStudent(@PathVariable Long id) {

			Optional<Student> std= studentRepo.findById(id);
			if(std.isPresent()) {
				studentRepo.deleteById(id);
```

```
                }

            }

}
```

7. SemesterController.java

```
package np.com.nast.noticeboard.controller;

import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/semesters")
public class SemesterController {
	@Autowired
	private SemesterRepository semRepo;

	//create
	@PostMapping
	public Semester createSemester(@RequestBody Semester semester) {
		return semRepo.save(semester);
	}

	@GetMapping
	public List<Semester> findAllSemester(){
		return semRepo.findAll();
	}

	@GetMapping("/{id}")
	public Optional<Semester> findSemester(@PathVariable Long id){
		return semRepo.findById(id);
	}

	@PutMapping("/{id}")
	public Semester createSemester(@RequestBody Semester semester,@PathVariable
Long id) {
		if(id==semester.getId())
			return semRepo.save(semester);
		return null;

	}
	@DeleteMapping("/{id}")
		public void deleteSemester(@PathVariable Long id) {

			Optional<Semester> sm= semRepo.findById(id);
			if(sm.isPresent()) {
```

```
                semRepo.deleteById(id);
            }

        }

}
```

8. UserRepository.java

```
package np.com.nast.noticeboard.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import np.com.nast.noticeboard.entity.User;

@Repository

public interface UserRepository extends JpaRepository <User,Long> {

}
```

9. SubjectRepository.java

```
package np.com.nast.noticeboard.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import np.com.nast.noticeboard.entity.Subject;

@Repository

public interface SubjectRepository extends JpaRepository <Subject,Long> {

}
```

10.TeacherController.java

```
package np.com.nast.noticeboard.controller;
import java.util.List;
import java.util.Optional;
import np.com.nast.noticeboard.repository.TeacherRepository;

@RestController
@RequestMapping("/teachers")
public class TeacherController {
```

```
		@Autowired
	private TeacherRepository teacherRepo;

	//create
	@PostMapping
	public Teacher createTeacher(@RequestBody Teacher teacher) {
		return teacherRepo.save(teacher);
	}

	@GetMapping
	public List<Teacher> findAllTeacher(){
		return teacherRepo.findAll();
	}

	@GetMapping("/{id}")
	public Optional<Teacher> findTeacher(@PathVariable Long id){
		return teacherRepo.findById(id);
	}

	@PutMapping("/{id}")
	public Teacher createTeacher(@RequestBody Teacher teacher,@PathVariable
Long id) {
		if(id==teacher.getId())
			return teacherRepo.save(teacher);
		return null;

	}
	@DeleteMapping("/{id}")
		public void deleteTeacher(@PathVariable Long id) {

			Optional<Teacher> tech= teacherRepo.findById(id);
			if(tech.isPresent()) {
				teacherRepo.deleteById(id);
			}

		}

}
```