SYSTEM TESTING

1. SYSTEM TESTING

1.1 TYPES OF TESTING

Testing is the major quality measure technique employed during software development process. After the coding phase, computer programs are available that can be executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus, the aim of testing is to uncover requirements, design or coding errors in the program.

The basic types of testing are:

- Unit testing
- Integration testing
- Validation testing
- Output testing
- User Acceptance testing

UNIT TESTING

This is the first level of testing. In this different module are tested against the specification produced during the design of the modules. Unit testing is done for the verification of code produced during the coding of single program module in an isolated environment. Unit testing first focuses onthe modules independently of one another to locate errors.

INTEGRATION TESTING

After the modules are tested individually, they must be tested in combination with each other tobe sure that the interfaces are correct. This is known as integration testing. Hence, we consider interfacing of various modules. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

VALIDATION TESTING

Validation testing gives the final assurances that the software meets all functional, behavioural and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which the user expects. Validation refers to the process of using software in a live environment in order to find errors. If the password was given wrongly by customers then it shows the check password error. Then if the username and password are not typed correct then it shows check username and password error. In the field, medicine quantity if the customers type any character other than numbers then it displays a warning message to give only numbers.

OUTPUT TESTING

After performing the validation testing the next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in to two ways, one is on screen and another is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. If the user gives their correct username and password then it logins to the corresponding page.

USER ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes where required.

1.2 TYPES OF VALIDATIONS

A Validation control enables to validate an input and display an error message if necessary.

Validation types are given below

REQUIRED FIELD VALIDATION

The Required Field Validator is actually very simple, and yet very useful. One can use it to make sure that the user has entered something in a Text Box control. In every form required field validator is assigned to full fill all the specification.

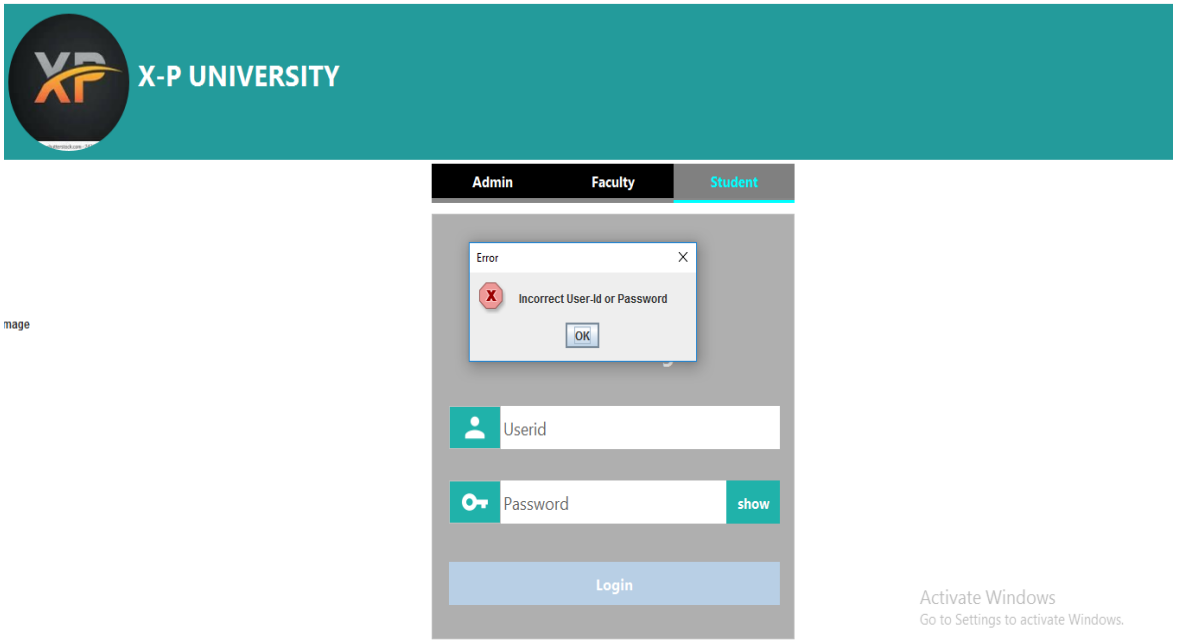
REGULAR EXPRESSION VALIDATION

Regular Expression Validator is one of the most useful validators, because it can be used to check the validity of any kind of string. In this project regular expression validator is assigned for email checking entries.

RANGE VALIDATION

The Range Validator does exactly what the name implies; it makes sure that the user input is within a specified range. It is used to validate numbers, strings and dates, which can make it useful in a bunch of cases. In this project, range validator is assigned for checking phone numbers.

1.3 ERROR MESSAGES



USER MANUAL

6 USER MANUAL

6.1 INSTALLATION MANUAL

- Download the required software.
- Unzip the files.
- Install all the software one by one.
- Install MYSQL.
- Install NetBeans.
- Unzip the JAVA file and Open it to the Apache NetBeans IDE.
- Open the Project
- Connect the database Server
- Right Click the Project and click the run option

6.2 OPERATIONAL MANUAL

1. Open the project and navigate to the home page.
2. Click on the "Login" option in the navigation bar to access the admin login.
3. Once logged in as an admin, you can manage users, courses, faculty, and student records.
4. If you are logged in as a student, you can view available courses, grades, and important announcements on the student dashboard.
5. Students can also register for courses by providing the necessary details and selecting their preferred classes.
6. Faculty members can log in to manage their courses, view student submissions, and access class schedules.
7. A dedicated registration page is available for different types of users, including students, faculty, and administrators.
8. Users can access career guidance resources that offer insights into potential career paths for freshers, final-year students, and those seeking professional advice.

SYSTEM IMPLEMENTATION

1. SYSTEM IMPLEMENTATION

SPECIAL FEATURES OF THE LANGUAGES

Java

Java is a widely-used, high-level programming language originally developed by Sun Microsystems and released in 1995. Designed with the principle of "Write Once, Run Anywhere," Java allows developers to create applications that can run on any device equipped with a Java Virtual Machine (JVM). It is an object-oriented language that emphasizes portability, security, and robustness, making it a popular choice for building web applications, mobile applications, and large-scale enterprise solutions.

The Java platform consists of the Java Development Kit (JDK), which includes tools for developing Java applications, and the Java Runtime Environment (JRE), which allows Java programs to run. Java's syntax is similar to that of C and C++, which makes it familiar to many developers.

UNIQUE FEATURES OF JAVA

Object-Oriented Programming: Java is built on the principles of object-oriented programming (OOP), which promotes code reuse and modularity. This approach allows developers to create classes and objects that encapsulate data and behavior, facilitating easier maintenance and scalability.

Platform Independence: Java's "Write Once, Run Anywhere" capability is made possible by the JVM, which allows Java applications to run on any operating system without modification. This feature greatly enhances cross-platform compatibility and reduces development costs.

Rich Standard Library: Java comes with a comprehensive standard library that provides a wide range of pre-built classes and functions for tasks such as networking, file handling, and data manipulation. This extensive library simplifies development and accelerates the creation of applications.

Strongly Typed Language: Unlike loosely typed languages, Java requires explicit declaration of data types. This feature enhances code clarity and reduces errors, as many type-related issues can be caught during compilation rather than at runtime.

Automatic Memory Management: Java uses a garbage collection system to automatically manage memory. This means that developers do not need to manually de allocate memory, reducing the risk of memory leaks and improving overall application stability.

Multithreading Support: Java provides built-in support for multithreading, allowing developers to create applications that can perform multiple tasks simultaneously. This feature is particularly useful for developing responsive applications that require concurrent processing.

Security Features: Java incorporates several security features, including a robust security manager and a built-in sandbox for running untrusted code. This makes Java a secure choice for developing applications that handle sensitive data.

Active Community and Ecosystem: Java has a vast and active community of developers and contributors. This community support translates into a wealth of resources, libraries, and frameworks, such as Spring and Hibernate, that enhance Java's functionality and ease of use.

Error Handling with Exceptions: Java uses a robust exception handling mechanism that allows developers to manage errors gracefully. This feature enables applications to recover from unexpected situations without crashing.

Integrated Development Environments (IDEs): Java developers benefit from a range of powerful IDEs, such as IntelliJ IDEA, Eclipse, and NetBeans, which provide tools for code completion, debugging, and project management, enhancing productivity and code quality.

Cross-Platform Compatibility: Java applications can run on various platforms, including Windows, macOS, and Linux, without the need for platform-specific adjustments. This portability simplifies deployment and maintenance.

Java Virtual Machine (JVM): The JVM is a key component of Java's architecture, enabling the execution of Java bytecode on any platform. It also optimizes performance through Just-In-Time (JIT) compilation.

Framework Support: Java boasts a rich ecosystem of frameworks that facilitate rapid application development. Frameworks like Spring, JavaServer Faces (JSF), and Play Framework simplify the creation of web applications and microservices.

In summary, Java is a versatile and powerful programming language that is well-suited for a wide range of applications, from web development to enterprise solutions, making it a valuable skill for any developer.

MySQL

MySql is an open source relational database management system (RDMBS). It's name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for structured Query Language.

MySql is free and open-source software under the terms of the GNU general public license, and is also available under a variety of proprietary licenses. MySql was owned and sponsored by the Swedish company MySql AB, which was bought by sun Microsystems. In 2010 when oracle acquired sun, Widenius forked the open-source MySql project to create MariaDB.

MySql is a component of the lamp web application software stack, which is an acronym for Linux, Apache, MySql, and Perl. MySql is used by many database-driven web applications, including drupal, joomla, phpBB, and Word Press. MySql is also used by many popular websites, including Google, face book, Twitter and You Tube. The main features of MySql includes

Easy to use: MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

It is secure: MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL

Client/ Server Architecture: MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.

Free to download: MySQL is free to use so that we can download it from MySQL official website without any cost.

Compatible on many operating systems: MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network orthe Internet).

FUTURE ENHANCEMENT

8. FUTURE ENHANCEMENT

The project "UNIVERSITY MANAGEMENT SYSTEM" can have the further updates.

1. Fees Payment Mode Enabling
2. Chat the students to faculties using the Application

CONCLUSION

9. CONCLUSION

The **"University Management System"** is an innovative project developed using Java and MySQL technologies that streamlines various administrative processes within a university setting. This platform provides a comprehensive solution for managing student information, course registrations, faculty details, and academic records, significantly enhancing the overall efficiency of university operations.

In the existing management systems, many processes are handled manually, leading to inefficiencies and delays in updates. Key challenges include difficulties in searching, sorting, and updating student data, as well as the lack of effective communication methods for disseminating information to students and faculty.

Our system addresses these issues by offering a user-friendly interface that allows students to easily access information about courses, grades, and announcements. Additionally, it enables faculty to manage their classes and student data effectively, while also facilitating administrative tasks such as enrollment and reporting.

With a focus on security and scalability, the project ensures data privacy and protection through robust authentication and authorization measures. Overall, this system provides a convenient and efficient way to manage university operations, improve communication, and support better decision-making based on comprehensive data analysis.

BIBLIOGRAPHY

10. BIBLIOGRAPHY

BOOK REFERENCES

- **Herbert Schildt**, "Java: The Complete Reference", Eleventh Edition, McGraw-Hill, 2019
- **Joshua Bloch**, "Effective Java", Third Edition, Addison-Wesley, 2018
- **Robert C. Martin**, "Clean Code: A Handbook of Agile Software Craftsmanship", Prentice Hall, 2008
- **Bruce Eckel**, "Thinking in Java", Fourth Edition, Prentice Hall, 2006
- **Martin Fowler**, "Refactoring: Improving the Design of Existing Code", Second Edition, Addison- Wesley, 2018

WEBSITE REFERENCES

• www.tutorialspoint.com

• www.stackoverflow.com

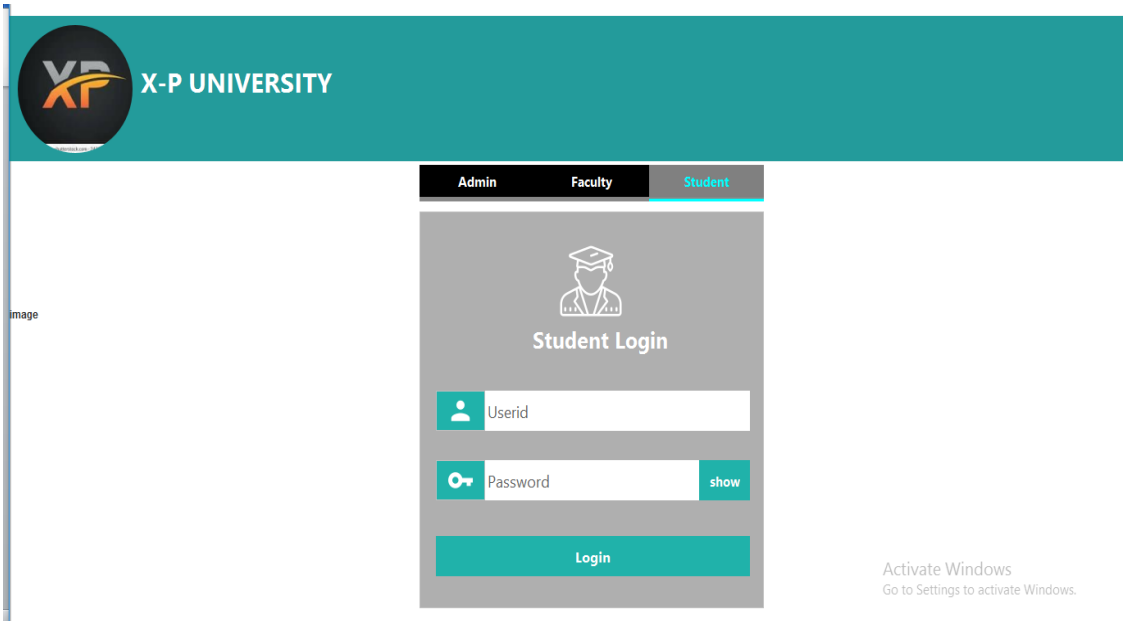
• www.w3schools.com

APPENDIX

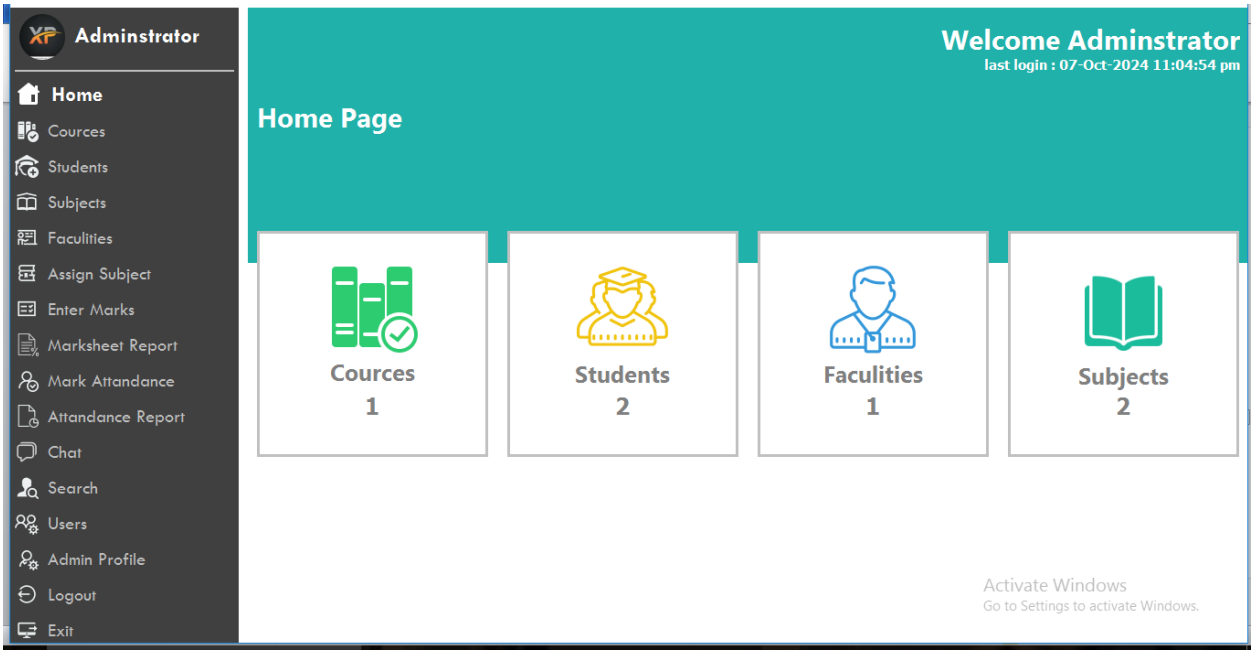
11. APPENDIX

11.1 SAMPLE SCREEN LAYOUTS

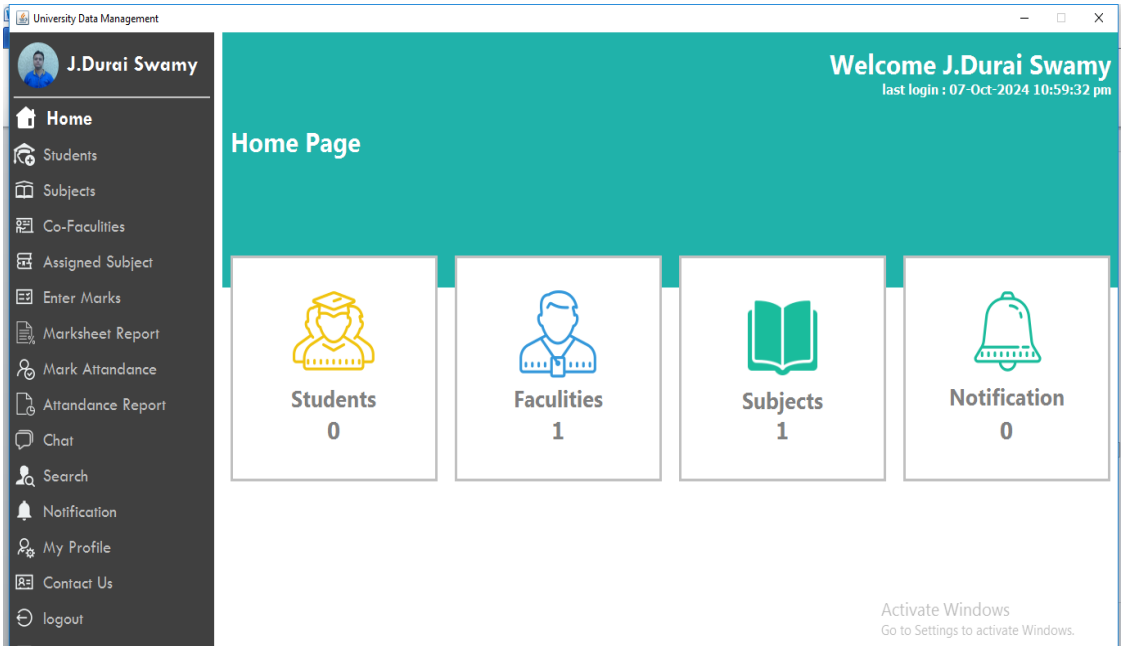
Student Login



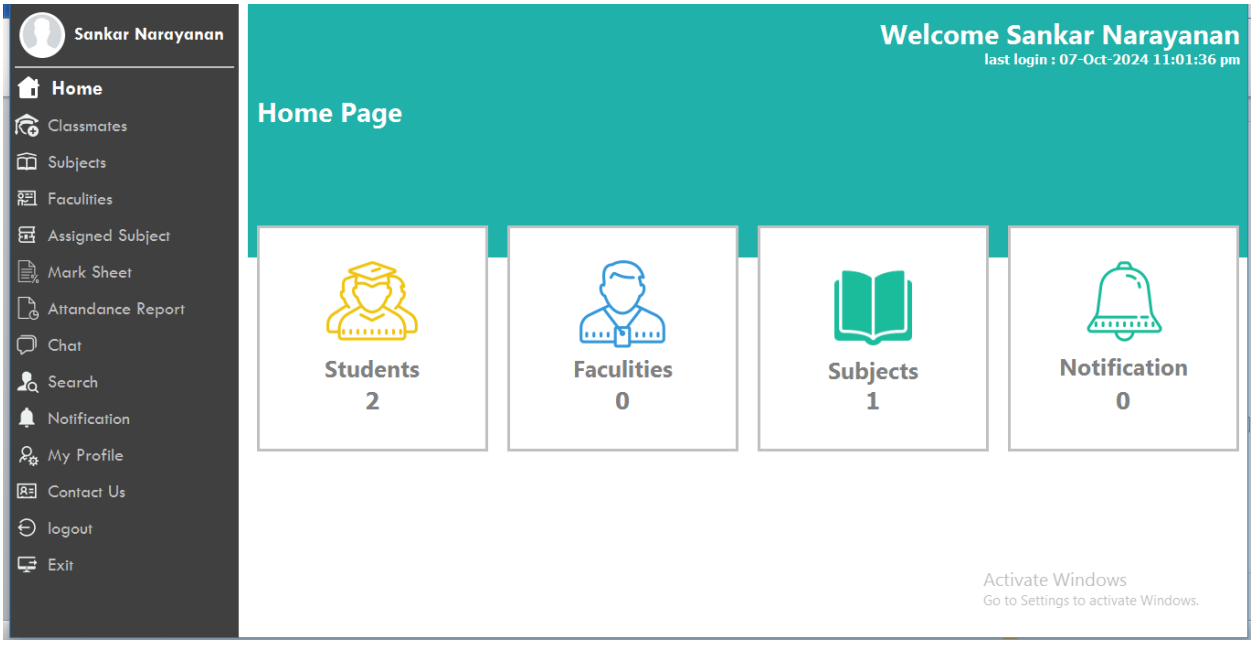
Admin Dashboard



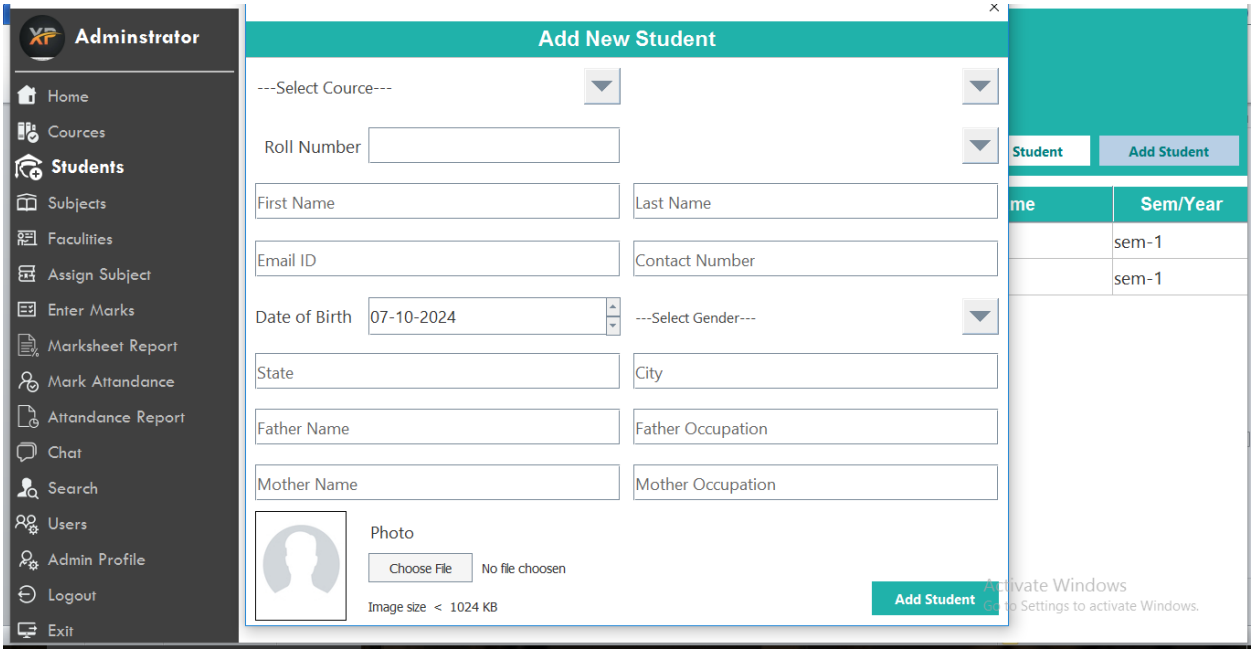
Faculty Dashboard

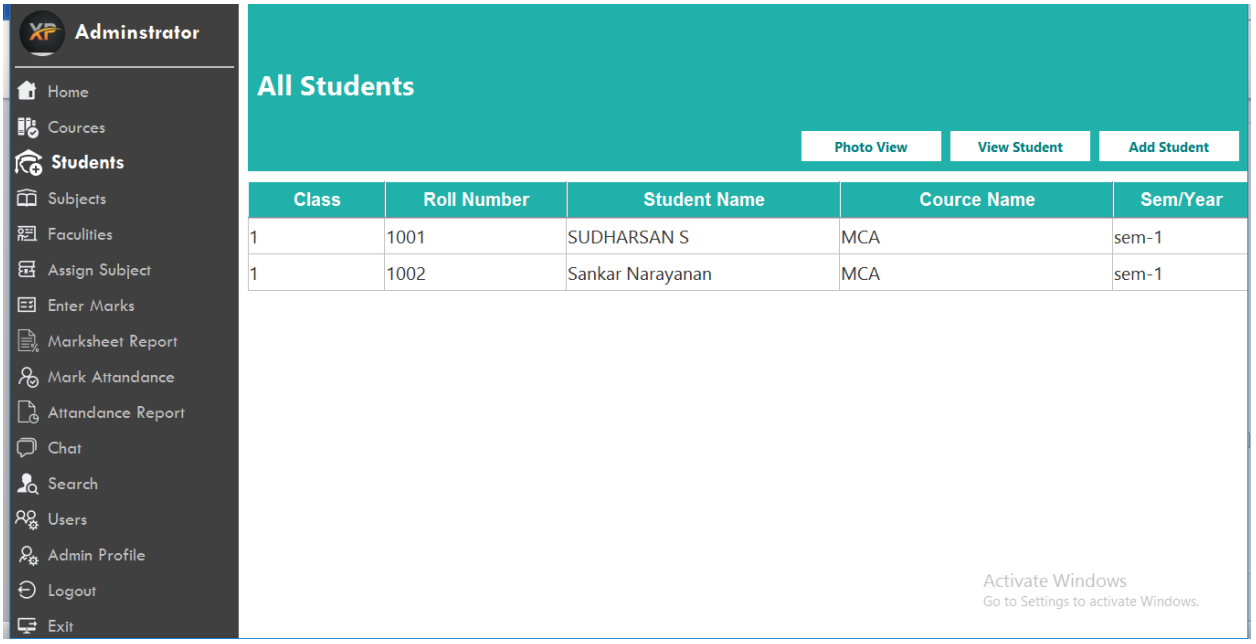


Student Dashboard

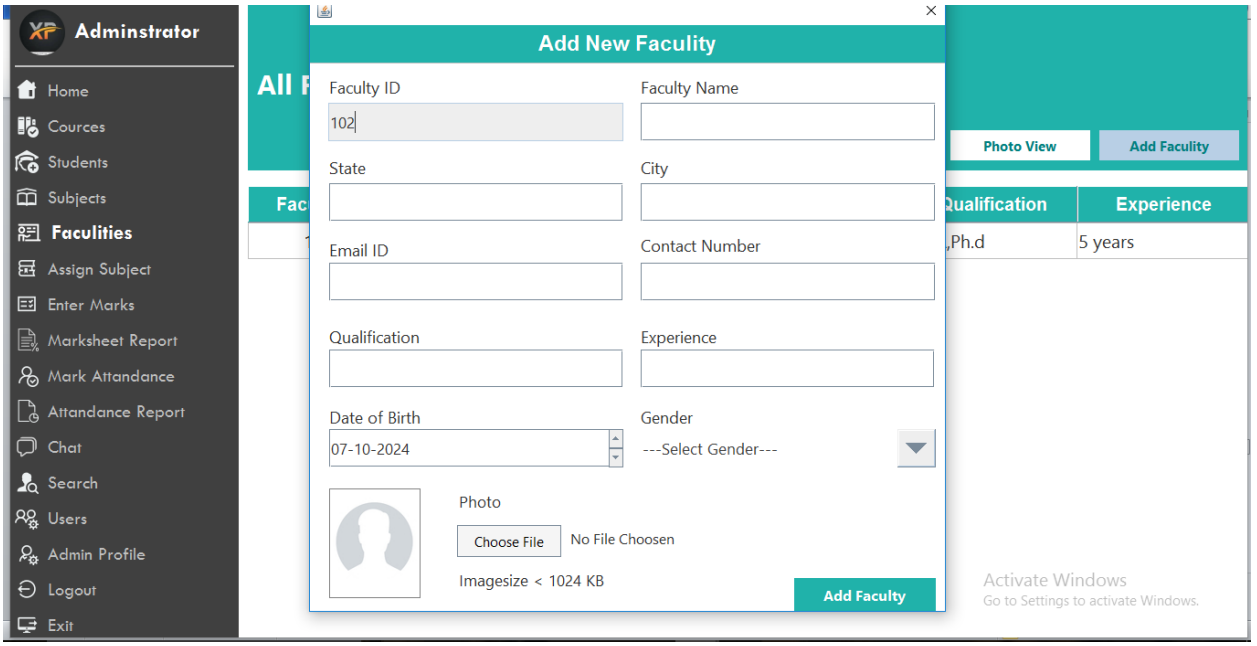


Add Student

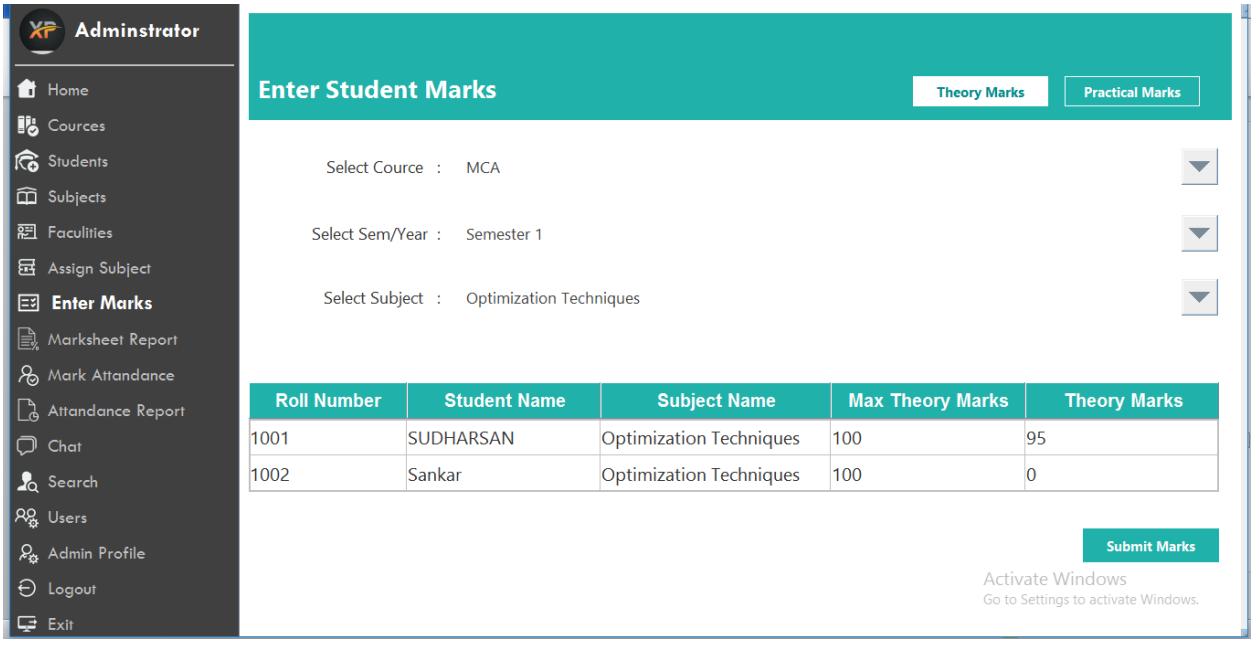

Adminstrator
Home
Cources
Students
Subjects
Faculities
Assign Subject
Enter Marks
Marksheet Report
Mark Attandance
Attandance Report
Chat
Search
Users
Admin Profile
Logout
Exit
Add New Student
---Select Cource---
Roll Number
First Name
Last Name
Email ID
Contact Number
Date of Birth
07-10-2024
---Select Gender---
State
City
Father Name
Father Occupation
Mother Name
Mother Occupation
Photo
Choose File
No file choosen
Image size < 1024 KB
Add Student
Student
Add Student
me
Sem/Year
sem-1
sem-1
Activate Windows
Go to Settings to activate Windows.


Adminstrator
Home
Cources
Students
Subjects
Faculities
Assign Subject
Enter Marks
Marksheet Report
Mark Attandance
Attandance Report
Chat
Search
Users
Admin Profile
Logout
Exit
All Students
Photo View
View Student
Add Student
Class
Roll Number
Student Name
Cource Name
Sem/Year
1
1001
SUDHARSAN S
MCA
sem-1
1
1002
Sankar Narayanan
MCA
sem-1
Activate Windows
Go to Settings to activate Windows.

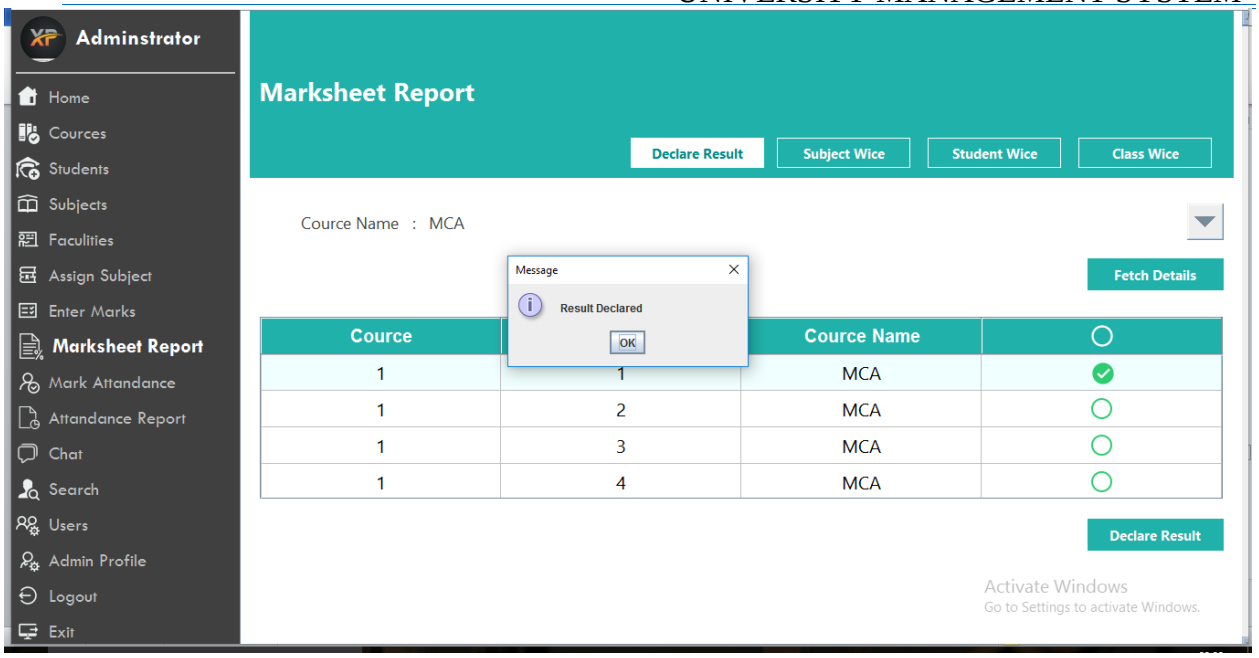
Add Faculty



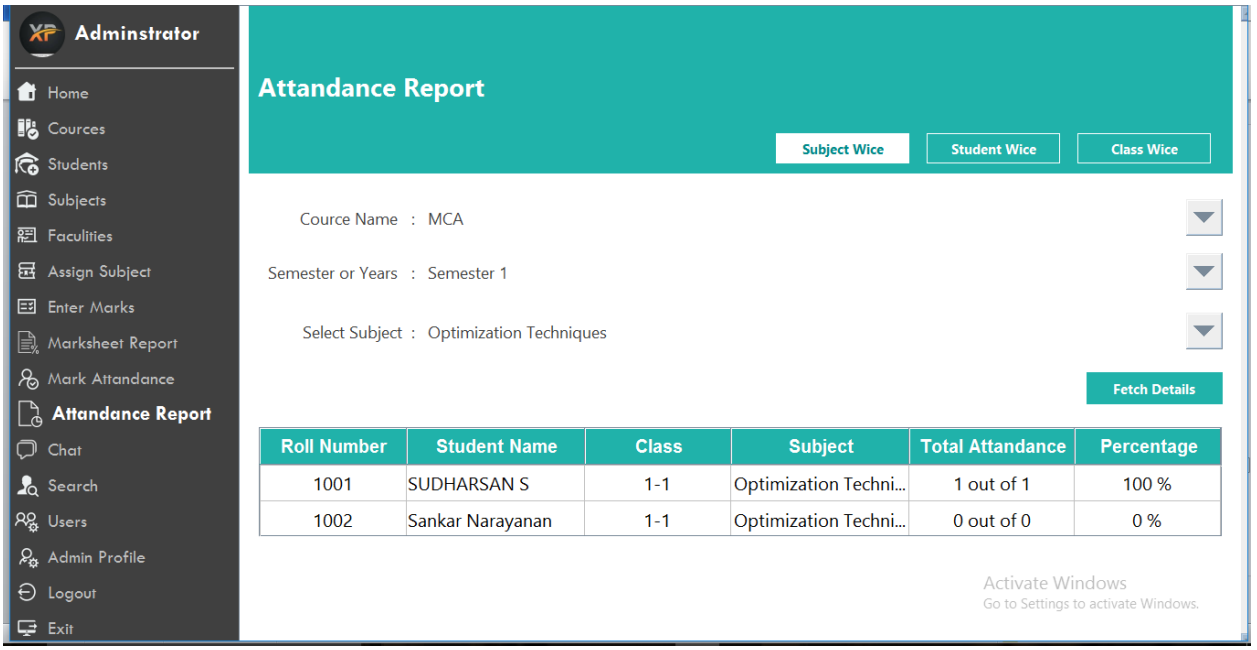
Mark Entering



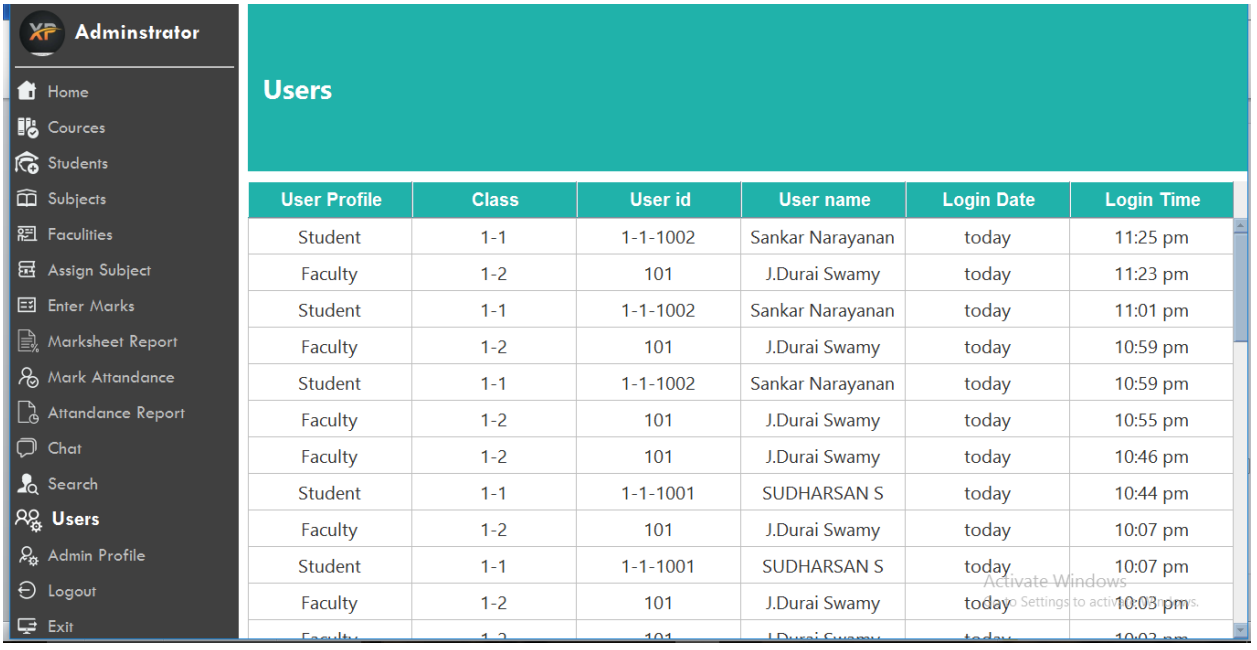
Declared Result


Adminstrator
Home
Cources
Students
Subjects
Faculities
Assign Subject
Enter Marks
Marksheet Report
Mark Attandance
Attandance Report
Chat
Search
Users
Admin Profile
Logout
Exit
Marksheet Report
Declare Result
Subject Wice
Student Wice
Class Wice
Cource Name : MCA
Message
Result Declared
OK
Fetch Details
Cource
Cource Name
1 1 MCA
1 2 MCA
1 3 MCA
1 4 MCA
Declare Result
Activate Windows
Go to Settings to activate Windows.

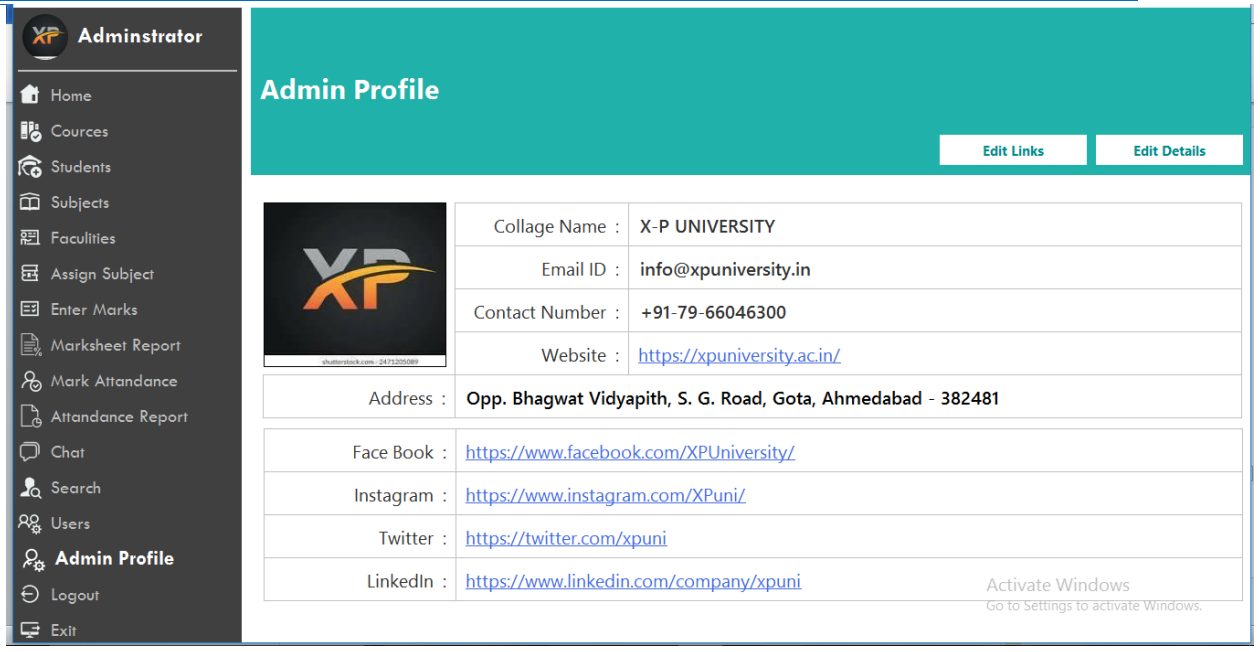
Attendance Report



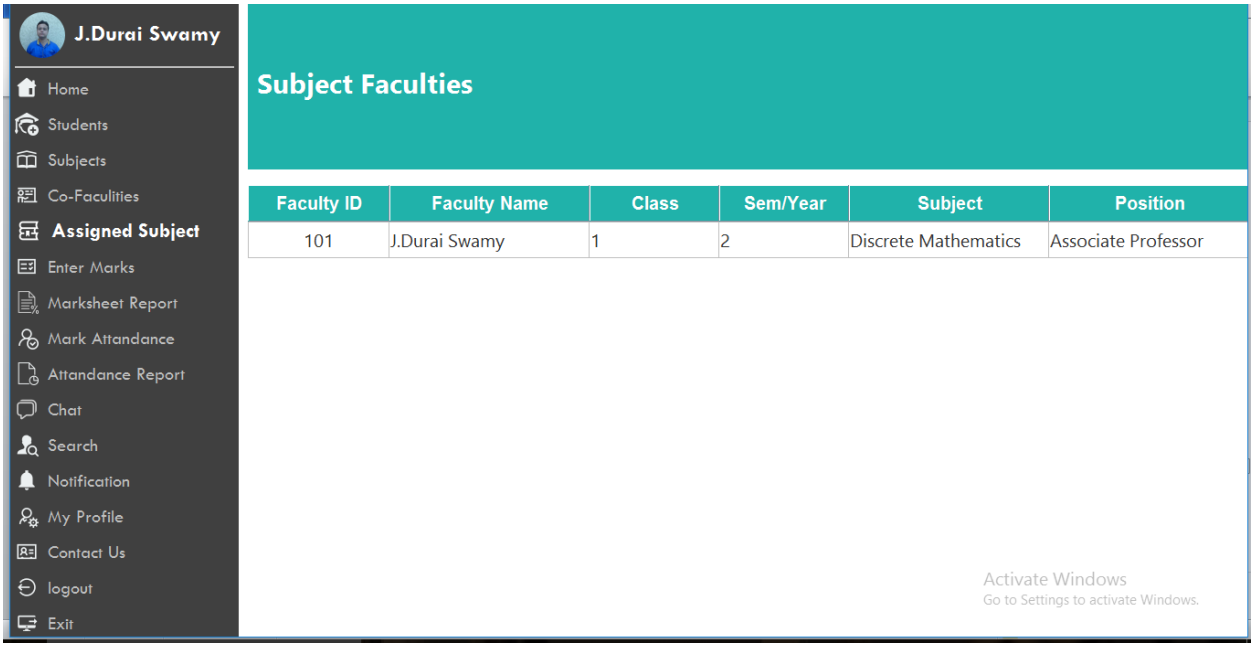
Users Login



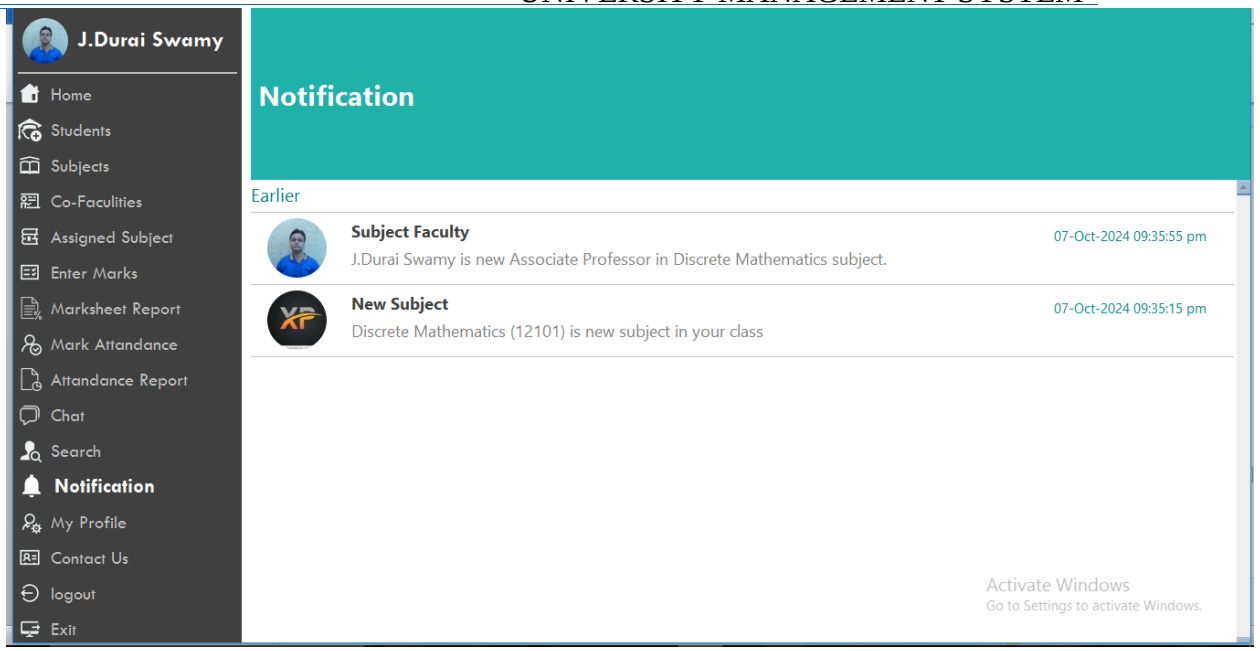
User Profile



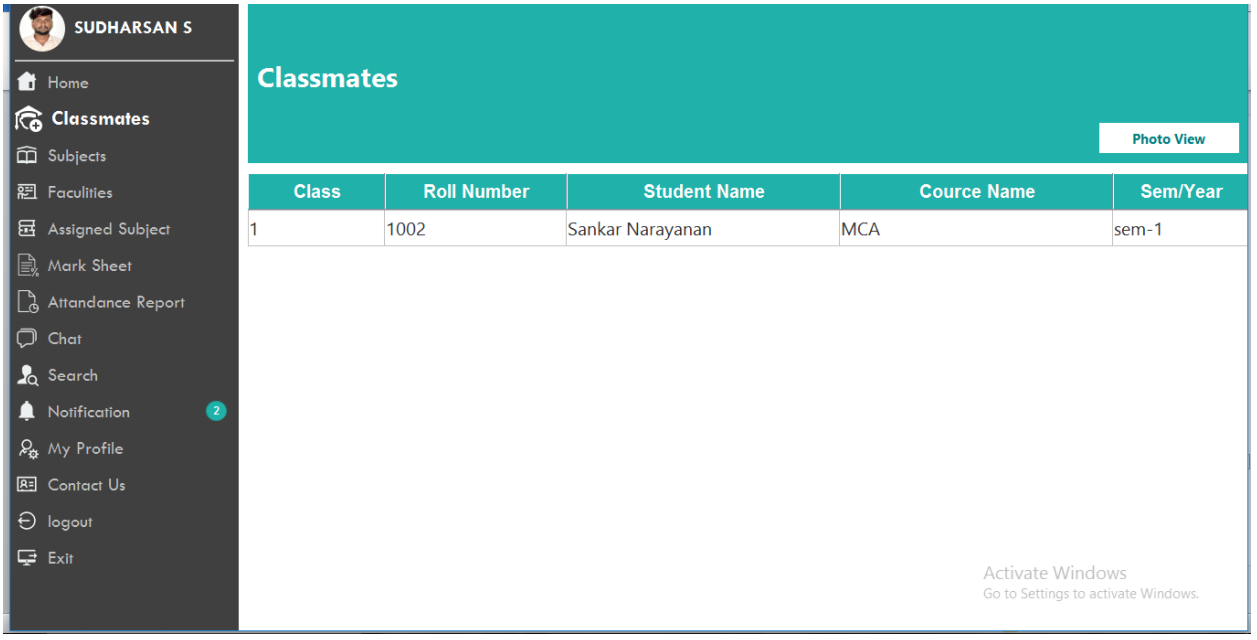
Assigned Subject



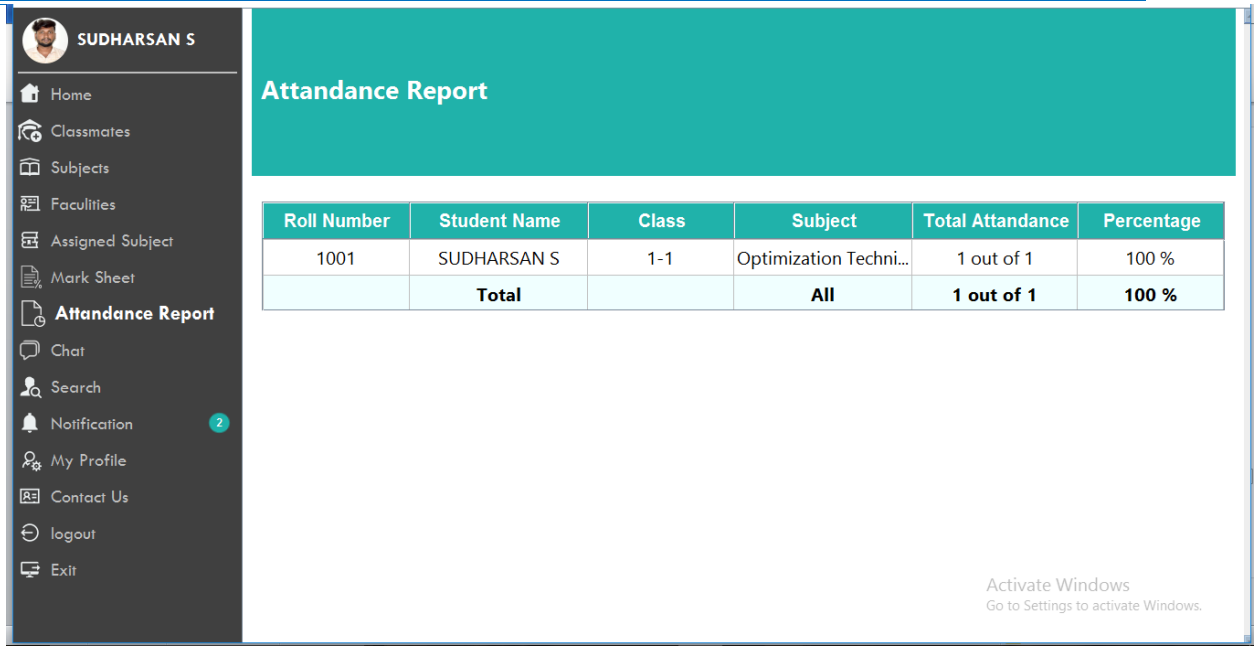
Notification Faculty



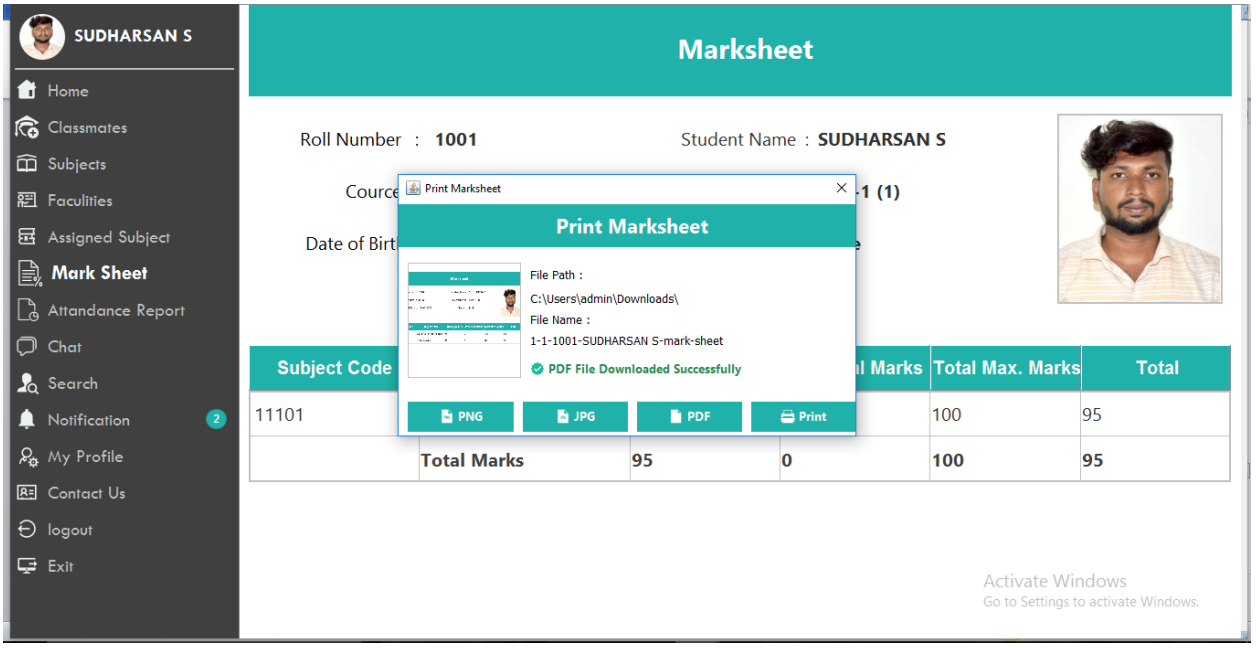
Class Mates View



Attendance Report



Mark Sheet Download



11.2 SAMPLE CODINGS

Admin:

```
package universityapplication.admin;

import java.awt.Image;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.IOException;

import javax.imageio.ImageIO;

import universityapplication.common.ImageUtil;

/*
 * Title : Admin.java
 * Created by : Sudharsan
 * Purpose : To bind all data of admin
 *
 */

public class Admin
{
private String website;
private String contactnumber;
private String emailid;
private String collagename;
private String password;
private Image logoimage;
private String facebook;
private String instagram;
private String twitter;
private String lastlogin;
private String linkedin;
private String address;
private boolean isactive=false;
public void setWebsite(String website)
{
	this.website=website;
}

public void setContactNumber(String contactnumber)
{
	this.contactnumber=contactnumber;
}
public void setEmailId(String emailid)
{
	this.emailid=emailid;
}
public void setCollageName(String collagename)
{
	this.collagename=collagename;
}
public void setPassword(String password)
{
	this.password=password;
}
```

```
public void setProfilePic(byte[] imagedata)
{
        this.logoimage=Toolkit.getDefaultToolkit().createImage(imagedata);
}
public void setProfilePic(Image profilepic)
{
        this.logoimage=profilepic;
}
public void setFaceBookLink(String facebooklink)
{
        this.facebook=facebooklink;
}
public void setInstagramLink(String instagramlink)
{
        this.instagram=instagramlink;
}
public void setLinkedinLink(String linkedinlink)
{
        this.linkedin=linkedinlink;
}
public void setTwitterLink(String twitterlink)
{
        this.twitter=twitterlink;
}
public void setLastLogin(String lastlogin)
{
        this.lastlogin=lastlogin;
}
public void setActiveStatus(boolean isactive)
{
        this.isactive=isactive;
}
public void setAddress(String address)
{
        this.address=address;
}
public String getWebsite()
{
        return website;
}
public String getContactNumber()
{
        return contactnumber;
}
public String getEmailId()
{
        return emailid;
}
public String getCollageName()
{
        return collagename;
}
public String getPassword()
{
        return password;
}
public boolean getActiveStatus()
{
        return isactive;
}
public String getFacebookLink()
```

```
	{
		return facebook;
	}
	public String getInstagramLink()
	{
		return instagram;
	}
	public String getTwitterLink()
	{
		return twitter;
	}
	public String getLinkedinLink()
	{
		return linkedin;
	}
	public Image getProfilePic()
	{
		return logoimage;
	}
	public byte[] getProfilePicInBytes()
	{
		ByteArrayOutputStream imagedata=new ByteArrayOutputStream();
		try {
			ImageIO.write(ImageUtil.toBufferedImage(logoimage), "jpg", imagedata);
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return imagedata.toByteArray();

	}

	public Image getProfilePic(int width,int height)
	{
		return logoimage.getScaledInstance(width, height, Image.SCALE_SMOOTH);
	}
	public BufferedImage getRoundedProfilePic(int width,int height,int radius)
	{
		return
ImageUtil.makeRoundedCorner(ImageUtil.toBufferedImage(logoimage.getScaledInstance(width,
height, Image.SCALE_SMOOTH)), radius);
	}
	public String getLastLogin()
	{
		return lastlogin;
	}
	public String getAddress()
			{
		return address;
	}


	}
```

Student:

```
package universityapplication.student;

import universityapplication.common.Person;
import universityapplication.common.TimeUtil;
import universityapplication.cource.CourceData;

public class Student extends Person{

	private long rollnumber;
	private String optionalsubject;
	private String firstname;
	private String lastname;
	private String fathername;
	private String fatheroccupation;
	private String mothername;
	private String motheroccupation;
	private String userid;
	private String admissiondate;


	public void setAdmissionDate(String admissiondate)
	{
		this.admissiondate=admissiondate;
	}
	public void setRollNumber(long rollnumber)
	{
		this.rollnumber=rollnumber;
	}
	public void setOptionalSubject(String optionalsubject)
	{
		this.optionalsubject=optionalsubject;
	}
	public void setFirstName(String firstname)
	{
		this.firstname=firstname;
	}
	public void setLastName(String lastname)
	{
		this.lastname=lastname;
	}
	public void setFatherName(String fathername)
	{
		this.fathername=fathername;
	}
	public void setFatherOccupation(String fatheroccupation)
	{
		this.fatheroccupation=fatheroccupation;
	}
	public void setMotherName(String mothername)
	{
		this.mothername=mothername;
	}
```

```
public void setMotherOccupation(String motheroccupation)
{
	this.motheroccupation=motheroccupation;
}
public void setUserId(String userid)
{
	this.userid=userid;
}
public String getCourceName()
{
	return new CourceData().getcourcename(this.getCourceCode());
}
public String getFullName()
{
	return firstname+" "+lastname;
}
public String getFirstName()
{
	return firstname;
}
public String getLastName()
{
	return lastname;
}
public long getRollNumber()
{
	return rollnumber;
}
public String getOptionalSubject()
{
	return optionalsubject;
}
public String getAdmissionDate()
{
	return admissiondate;

}
public String generateAdmissionDate()
{
	admissiondate=TimeUtil.getCurrentTime();
	return admissiondate;
}
public String getFatherName()
{
	return fathername;
}
public String getMotherName()
{
	return mothername;
}
public String getFatherOccupation()
{
	return fatheroccupation;
}
public String getMotherOccupation()
```

```
		{
			return motheroccupation;
		}
		public String getUserId()
		{
			return userid;
		}
		public String generateUserId()
		{
			// TODO Auto-generated method stub
			userid=getCourceCode()+"-"+getSemorYear()+"-"+rollnumber;
			return userid;
		}

}
```