

A
Project Report
on
“Grievance Redressal College Portal”

Submitted to



Savitribai Phule Pune University, Pune
In Partial Fulfillment of
Master of Computer Application
(MCA - I, Sem - II)

Submitted by
Mr. Sachin Somnath Sargar

Under the Guidance of
Prof. Akash Raut

Through



Dr. D. Y. Patil Educational Federation's
Dr. D. Y. Patil Institute of Management and Entrepreneur
Development.
2024-2025



Dr. D. Y. PATIL EDUCATIONAL FEDERATION

Dr. D. Y. Patil Institute of Management and Entrepreneur Development

Approved by AICTE, Affiliated to SPPU Pune

Address: - Sr. No. 27/A/1/2C, Near Talegaon Railway Station, Towards HP Service Station, Varale, Talegaon, Pune, Maharashtra 410507. Website: dypatilef.com. dypatilimed.com

E-mail- director.dypimed@dypatilef.com

Dr. D. Y. Patil
Founder Dr. D Y Patil Group

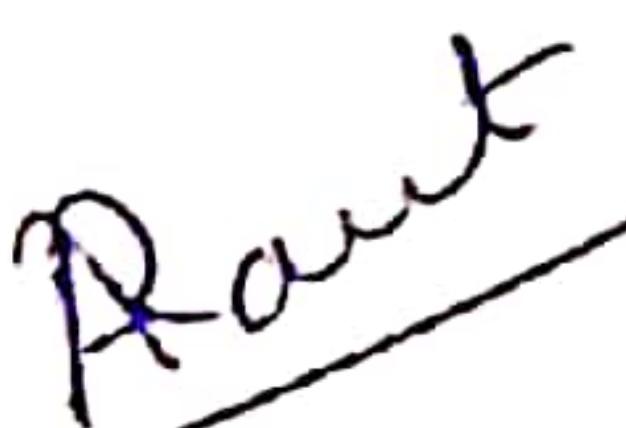
Dr. Sushant Patil
President

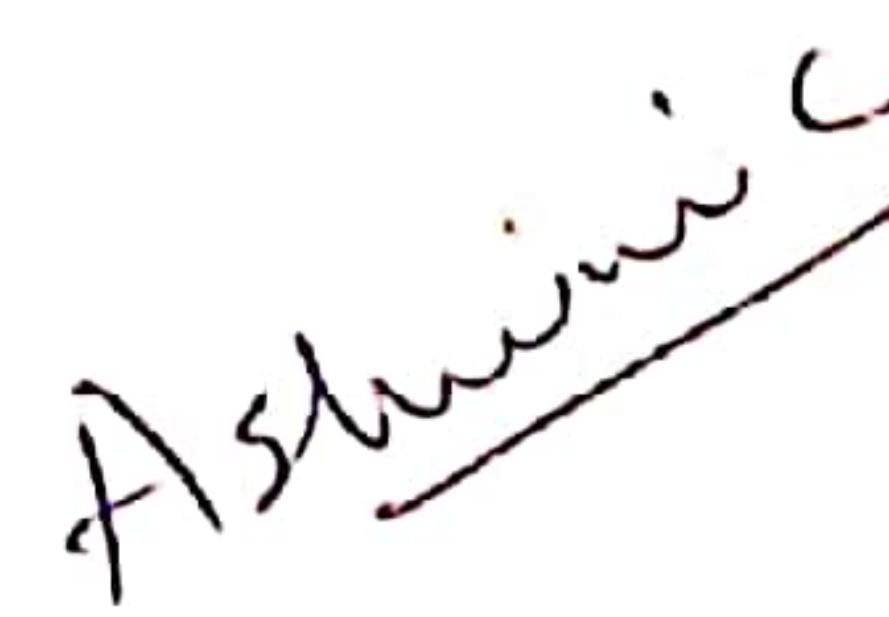
Dr. Meghana Bhilare
Director

CERTIFICATE

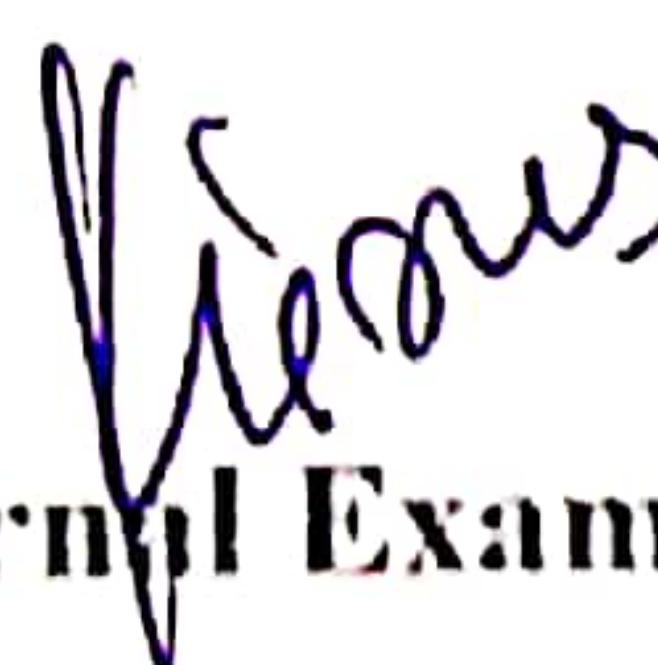
This is to Certify that Project entitled "**Grievance Redressal Collage Portal**" is submitted by **Mr. Sachin Somnath Sargar** has successfully Completed the Project under the guidance of **Prof. Akash Raut**.

This Project is submitted to the **Savitribai Phule Pune University** in Partial fulfilment of the requirement for the award of the degree of Master of Computer Application (MCA I Sem. II Batch 2024-2026).


Prof. Akash Raut
Project Guide

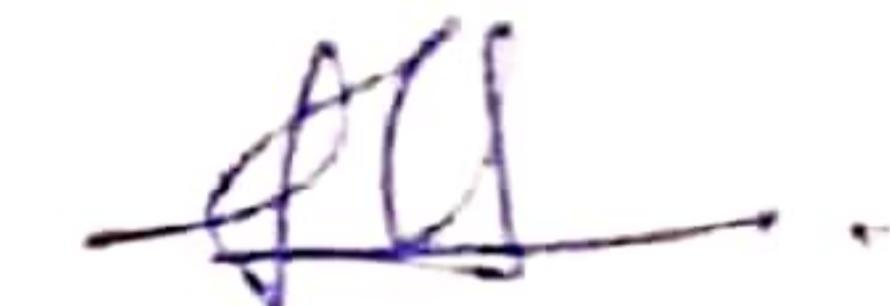

Dr. Ashwini Chavan
HoD


Dr. Meghana Bhilare
Director


Internal Examiner

DECLARATION

I, the undersigned hereby declare that the project entitled "**Grievance Redressal College Portal**", being submitted for the award of degree of **Master of Computer Application(MCA – I, Sem – II)** by me to **Dr. D. Y. Patil Institute of Management and Entrepreneur Development, Varale, Talegaon, Pune affiliated to Savitribai Phule Pune University, Pune** is the result of an independent work carried out under the guidance of **Prof. Akash Raut** is my original work . Further, I declare that this project has not been submitted to this or any institution for the award of any degree.



PLACE: *Varale*

Mr. Sachin Somnath Sargar

DATE: *12 - 4 - 2025*

ACKNOWLEDGEMENT

The project developed for the MCA was not possible without the persons and organizations that helped me in completing this. I am deeply grateful to all whose enthusiasm and energy transformed my vision of this study into reality.

I extend my sincere thanks to **Prof. Akash Raut** for making it easy to work in the Institute and providing me needed guidance throughout the project keeping it focused and on the track. I am thankful to him/her for the extended knowledge imparted to me during the course of project development.

I take this opportunity to thank our **HOD Dr. Ashwini Chavan** and our **Director Dr. Meghana Bhilare**, for encouragement and guidance throughout the progress of this report.

Mr. Sachin Somnath Sargar

INDEX

Chapter no.	Details	Page No.
1	Introduction	1
	1.1 Existing System and Need For System	2
	1.2 Scope of System	3
	1.3 Operating Environment – Hardware and Software	4
	1.4 Brief Description of Technology Used 1.4.1 Operating systems used (Windows or Unix) 1.4.2 RDBMS/No Sql used to build database (mysql/ oracle, Teradata, etc.)	5
2	Proposed System	
	2.1 Study of Similar Systems (If required research paper can be included)	7
	2.2 Feasibility Study	8
	2.3 Objectives of Proposed System	9
	2.4 Users of System	10
3	Analysis And Design	
	3.1 System Requirements (Functional and Non-Functional requirements)	11
	3.2 Entity Relationship Diagram (ERD)	12
	3.3 Use case Diagram	13
	3.4 Class Diagram	15
	3.5 Activity Diagram	16
	3.6 Deployment Diagram	18
	3.7 Data Structure	19
	3.8 Sample Screens (Screens must have valid data. All reports must have at-least 5 valid records.)	21
4	Coding	
	4.1 Coding Snippets	29
5	Testing	
	5.1 Test Case	33
6	Limitations of Proposed System	37
7	Proposed Enhancement	38
8	Conclusion	39
9	Bibliography	40
10	User Manual	41

1 .Introduction

College is a place where students come to learn, grow, and prepare for their future. While college life is often exciting, it's also a place where problems can come up. These problems might be related to studies, classrooms, hostels, staff behavior, exam issues, or even problems with facilities like the library or Wi-Fi. When students face such issues, they need a way to report them and get help.

In many colleges, if a student has a complaint, the usual method is to write a letter and give it to a teacher or office staff. Some students are even asked to meet with the head of the department or principal. This method is old, slow, and not very clear. Students don't always know if their complaint has been accepted or if someone is working on it. Many students feel ignored or frustrated.

To solve this problem, we created a digital solution called the **Grievance Redressal College Portal**.

This portal is like a website made just for students and college staff. Students can create an account, log in, and easily fill out a form to submit their complaint. They can also check the portal later to see what's happening with their complaint—whether it has been seen, is being worked on, or has been solved. No more running around or waiting for someone to respond.

But this portal is not just for students. It also helps the college administration. The admin panel allows staff members to log in and see all the student complaints in one place. They can read the complaints, reply with updates or solutions, and even post important messages or notices. This saves time for both students and the admin, and it keeps everything well organized.

By using this portal, communication becomes smoother. Students don't feel ignored anymore, and college staff can solve problems in a better way. It also helps build trust between students and the administration.

1.1 Existing System and Need for the System

In many educational institutions, especially colleges, students frequently face problems related to academics, infrastructure, hostel facilities, staff behavior, or general administration. However, the way these problems are currently reported and handled is mostly outdated and inefficient. Traditionally, if a student wants to raise a complaint or express a concern, they are required to write a letter and submit it physically to a teacher, head of the department, or principal. This paper-based approach is time-consuming, lacks transparency, and often fails to provide a satisfactory experience for the student. In many cases, students are not informed whether their complaint has been received or if any action has been taken. As a result, students often feel ignored, unimportant, and disconnected from the administration.

Moreover, the manual process does not provide any form of status tracking. Once a complaint is submitted, the student has no way of knowing if someone is working on it, whether it has been escalated to a higher authority, or when they can expect a resolution. There is also a high chance of complaints getting lost, misplaced, or delayed due to human errors. With no proper system to store and manage these complaints, it becomes difficult for administrators to keep track of past records or analyze recurring problems. This lack of structure results in delayed responses, unresolved issues, and general dissatisfaction among students.

In today's world, where most services have moved online, it is necessary for colleges to also adopt digital tools that make student services more effective. An online

grievance redressal system is the need of the hour. Such a system can provide a centralized platform where students can register and log in securely, submit their complaints through a user-friendly form, and check the status of their complaints in real time. This ensures that students are kept informed about the progress and feel that their voice matters. On the other hand, college administrators can view, manage, prioritize, and resolve complaints more efficiently using an admin dashboard. They can also maintain proper records of each complaint, track resolution timelines, and generate reports when required.

1.2 Scope of the System

The **Grievance Redressal College Portal** is a web-based platform developed to make the complaint process simple, fast, and transparent for students. Its main goal is to provide students with an easy way to report their issues and stay informed about what's happening with their complaints. Instead of writing letters or standing in lines, students can now register on the portal, log in securely, and submit their complaints using an online form. The system then allows them to check the status of their complaint anytime, so they always know whether it is being reviewed, in progress, or resolved. This gives students peace of mind and confidence that their problems are being taken seriously.

At the same time, the portal also supports the needs of the college administration. It gives admins a secure login and access to a dashboard where they can see all the complaints submitted by students. Admins can read each complaint, take appropriate action, update the status, and even send replies or updates back to the student. In

addition to handling complaints, administrators can use the portal to post important notices or announcements that are visible to all students. Notices can be grouped into categories such as "Regular Notices" or "Important Notices" to help students identify key updates at a glance.

The system ensures that all user data—whether it belongs to students or staff—is handled securely and responsibly. The portal also maintains records of complaints and responses, so nothing is lost and everything can be reviewed if needed in the future. Because the platform is web-based, it can be accessed from any device with internet access, including computers, laptops, tablets, and smartphones, making it convenient for all users.

In the future, this system can be expanded to handle even more features. For example, different departments can have their own access to manage subject-specific complaints. There can also be an escalation mechanism where unresolved issues automatically move to a higher authority. A feedback system can be added to allow students to rate the resolution process or suggest improvements. This portal can even be used as a reference or model for other colleges that want to modernize their grievance handling system. Overall, the system is scalable, flexible, and highly useful in building better communication between students and the administration.

1.3 Operating Environment - Hardware and Software

Hardware Requirements:

Server:

- ✓ Minimum 4-core CPU
- ✓ 8 GB RAM
- ✓ 100 GB SSD storage

Client Devices:

- ✓ Any internet-connected device such as a laptop, desktop, tablet, or smartphone
- ✓ Modern web browser support (e.g., Chrome, Firefox)

Software Requirements:**Operating System:**

- ✓ Windows 10 or 11 (for development)
- ✓ Linux – Ubuntu (preferred for deployment)

Backend Technology:

- ✓ Node.js with Express.js framework

Frontend Technology:

- ✓ React.js (for building user interfaces)

Development Tools:**Code Editor:**

- ✓ Visual Studio Code (VS Code)

API Testing Tool:

- ✓ Postman

Version Control System:

- ✓ Git

Database GUI:

- ✓ MongoDB Atlas

1.4 Brief Description of Technology Used

1.4.1 Operating Systems Used

The Grievance Redressal College Portal was developed using the Windows operating system, specifically Windows 10 and Windows 11. Windows was chosen due to its user-friendly interface, wide availability, and strong compatibility with essential development tools. Its graphical interface makes it easy for developers to interact with the system and perform tasks without relying heavily on command-line instructions, which helps speed up the development process, especially for students and beginners.

Windows supports all the key tools used in this project, including Visual Studio Code (VS Code) for writing and editing code, Postman for API testing, MongoDB and MongoDB Compass for database management, and Git for version control. These tools work seamlessly on Windows, making it an ideal environment for developing full-stack applications using the MERN (MongoDB, Express, React, Node.js) stack.

In addition, most developers are already familiar with the Windows operating system from their academic or personal use, which reduces the learning curve and allows them to focus more on the project itself. Windows also offers excellent driver and hardware support, ensuring that devices like USBs, printers, and external monitors work without compatibility issues during development and testing.

Furthermore, Windows has a large online support community and is backed by Microsoft's regular updates, which help in resolving any technical issues quickly. Throughout the development of this portal, all coding, database setup, API integration, and testing were done on Windows. Browsers like Chrome and Microsoft Edge on Windows were used to test the portal's responsiveness and functionality. Overall, Windows provided a reliable and efficient platform for building the Grievance Redressal College Portal from start to finish.

1.4.2 RDBMS/NoSQL Used

In this project, we are using **MongoDB**, a powerful and widely-used **NoSQL (Non-Relational Database)** system. Unlike traditional relational databases such as MySQL or Oracle, which store data in rows and columns inside fixed tables, MongoDB stores information in a more flexible format using **JSON-like documents**. This flexibility makes MongoDB highly suitable for modern web applications like our **Grievance Redressal College Portal**, where the structure of data can vary or evolve over time.

The reason for choosing MongoDB over a traditional RDBMS is based on several key factors. Firstly, MongoDB does not require a fixed schema, which means we can easily make changes to how data is stored without affecting the existing database structure. For example, if we later decide to add new details to a student's complaint such as feedback or a file attachment, we can simply include them in the document without needing to modify any tables or perform complex migrations.

Another major advantage is **performance**. MongoDB is optimized for **fast read and write operations**, which is important in a system where multiple students may be filing complaints and checking updates at the same time. It also supports **horizontal scaling**, allowing the application to handle increasing data loads efficiently by spreading the data across multiple servers if needed.

Additionally, MongoDB provides **powerful querying and indexing** features. This means we can quickly search for complaints by status, student ID, date, or other criteria without slowing down the application. In our portal, we use MongoDB to store all

essential information including student accounts, complaints, admin responses, and notices. The documents are organized in such a way that both students and admins can interact with the system in real-time, with immediate access to updates and responses.

Overall, MongoDB offers a robust, flexible, and scalable solution for managing the kinds of dynamic data we work with in this project. It enhances the user experience by ensuring speed and reliability, and it helps developers by simplifying data handling and reducing the time needed for future updates or improvements.

2: Proposed System

2.1 Study of Similar Systems

Observations from Existing Systems:

1. Many institutions have basic online complaint portals.
2. Most lack **real-time complaint tracking**, leading to confusion among students.
3. Poor **categorization of issues**, making it difficult for admins to manage and prioritize complaints.
4. **Unattractive or confusing user interfaces**, which affect usability and student engagement.
5. Absence of **role-based access control**, leading to security and privacy concerns.
6. Failure to **close the feedback loop**, so students don't know if or when their issue was resolved.

How Our Proposed System Is Better:

- ✓ **Real-time updates:** Students receive live status changes on their complaints.

- ✓ **Categorized complaint submission:** Issues are grouped (e.g., academic, hostel, facilities) for better management.
- ✓ **User-friendly interface:** Clean and simple UI for both students and admins.
- ✓ **Role-based access control:** Students and admins have different permissions and dashboards.

2.2 Feasibility Study

Technical Feasibility

- ✓ The system is developed using the **MERN stack**, which includes **MongoDB** (database), **Express.js** (backend framework), **React.js** (frontend library), and **Node.js** (runtime environment). This stack enables end-to-end development using JavaScript.
- ✓ All technologies used are **open-source, free**, and have large support communities. This makes them reliable and easy to troubleshoot.
- ✓ The MERN stack is **well-tested**, scalable, and used by many modern web applications, making it a technically solid choice.
- ✓ It supports **cross-platform development**, which means the portal can be accessed on any operating system and device with a modern browser.

Operational Feasibility

- ✓ The portal has been designed keeping in mind the users— **students and administrators**— who may or may not have technical backgrounds.
- ✓ The **user interface (UI)** is clean, simple, and intuitive, making it easy to navigate without needing technical knowledge.
- ✓ Only **minimal training** or orientation is required, as all functions like complaint submission, status tracking, and admin responses are straightforward and guided.
- ✓ The system is **responsive** and can be used comfortably on desktops, laptops, tablets, and smartphones.

Economic Feasibility

- ✓ One of the major advantages of this system is its **cost-effectiveness**. Since all core technologies (MongoDB, Express, React, Node) are **completely free and open-source**, there are **no licensing fees** involved.
- ✓ Development can be done using free tools like **VS Code**, **Postman**, and **GitHub**.

2.3 Objectives of the Proposed System

- ✓ **Centralized Platform:**

To develop a unified online platform where all student grievances can be submitted, viewed, and managed in one place. This helps maintain transparency and ensures that no complaint is overlooked.

- ✓ **Easy Complaint Submission and Tracking:**

To provide students with a simple and user-friendly interface where they can easily file complaints and **track the real-time status** of their submissions at any time.

- ✓ **Efficient Complaint Resolution for Admins:**

To equip administrators with tools to view, manage, and resolve complaints efficiently. Admins can filter complaints by category, urgency, or date to respond in a timely manner.

- ✓ **Notice Management Feature:**

To include a module where administrators can **post important notices or announcements**, ensuring students are kept informed about updates, events, or system-related communications.

- ✓ **Deadline Management Feature:**

To include a module where administrators can **post important deadline or announcements**, ensuring students are kept informed about updates, campus drive.

- ✓ **Secure Data Handling and Access Control:**

To implement proper **authentication and role-based access control** so that sensitive student data is protected and only authorized users can access complaint-related information.

2.4 Users of the System

Students

Students are the primary users who will use the portal to raise concerns and stay informed. Their key features include:

- ✓ **Register:** Sign up on the portal using their basic information and credentials.
- ✓ **Login:** Access their dashboard securely using username and password.
- ✓ **Submit Complaints:** File grievances under different categories (e.g., academics, hostel, facilities).
- ✓ **Track Complaint Status:** Monitor the progress of their submitted complaints in real time (e.g., pending, in Progress, resolved).
- ✓ **View Notices:** Read **general** as well as **important notices** posted by the admin, such as college updates, holiday announcements, or event notifications.
- ✓ **Receive Deadline Alerts:** Get notified about **important deadlines** related to complaint resolutions or submission timelines.

Administrators (Admins)

Admins are responsible for managing the portal, resolving complaints, and maintaining communication with students. Their key functionalities include:

- ✓ **Login:** Securely access the admin dashboard using assigned credentials.
- ✓ **View Student Details:** Access the list of registered students and their complaint history.
- ✓ **Manage Complaints:** View, categorize, prioritize, and take action on complaints submitted by students.
- ✓ **Resolve Issues:** Provide written resolutions, mark complaints as resolved, and close tickets.

- ✓ **Post Notices:** Share announcements categorized as:
- ✓ **General Notices** (e.g., exam schedules, form deadlines)
- ✓ **Important Notices** (e.g., urgent updates, emergency changes, policy notices)
- ✓ **Set Deadlines:** Add and manage **resolution deadlines** or **submission cut-offs** to improve response times and maintain accountability.

3: Analysis and Design

3.1 System Requirements

Functional Requirements:

- ✓ **User Registration and Authentication:**
The system allows both students and administrators to securely register and log in using a unique email and password. Authentication ensures only authorized users access the portal.
- ✓ **Submit, View, and Track Complaints:**
Students can submit new grievances, view previously filed complaints, and check the status of each complaint (e.g., pending, in-progress, resolved) through their dashboard.
- ✓ **Admin Dashboard:**
Administrators can access a dedicated dashboard to manage student users, view complaints, categorize them, assign deadlines, and monitor progress.
- ✓ **Complaint Resolution Functionality:**
Admins can review each complaint, take appropriate action, respond with resolutions, and mark complaints as resolved once they are addressed.
- ✓ **Add and View Notices:**
Administrators have the ability to post important and general notices, which are instantly viewable by students on their dashboard. This helps keep students updated on key deadlines, announcements, and policies.

Non-Functional Requirements:

- ✓ **Security:**
The portal must enforce strong authentication and role-based access control to protect user data and prevent unauthorized access.

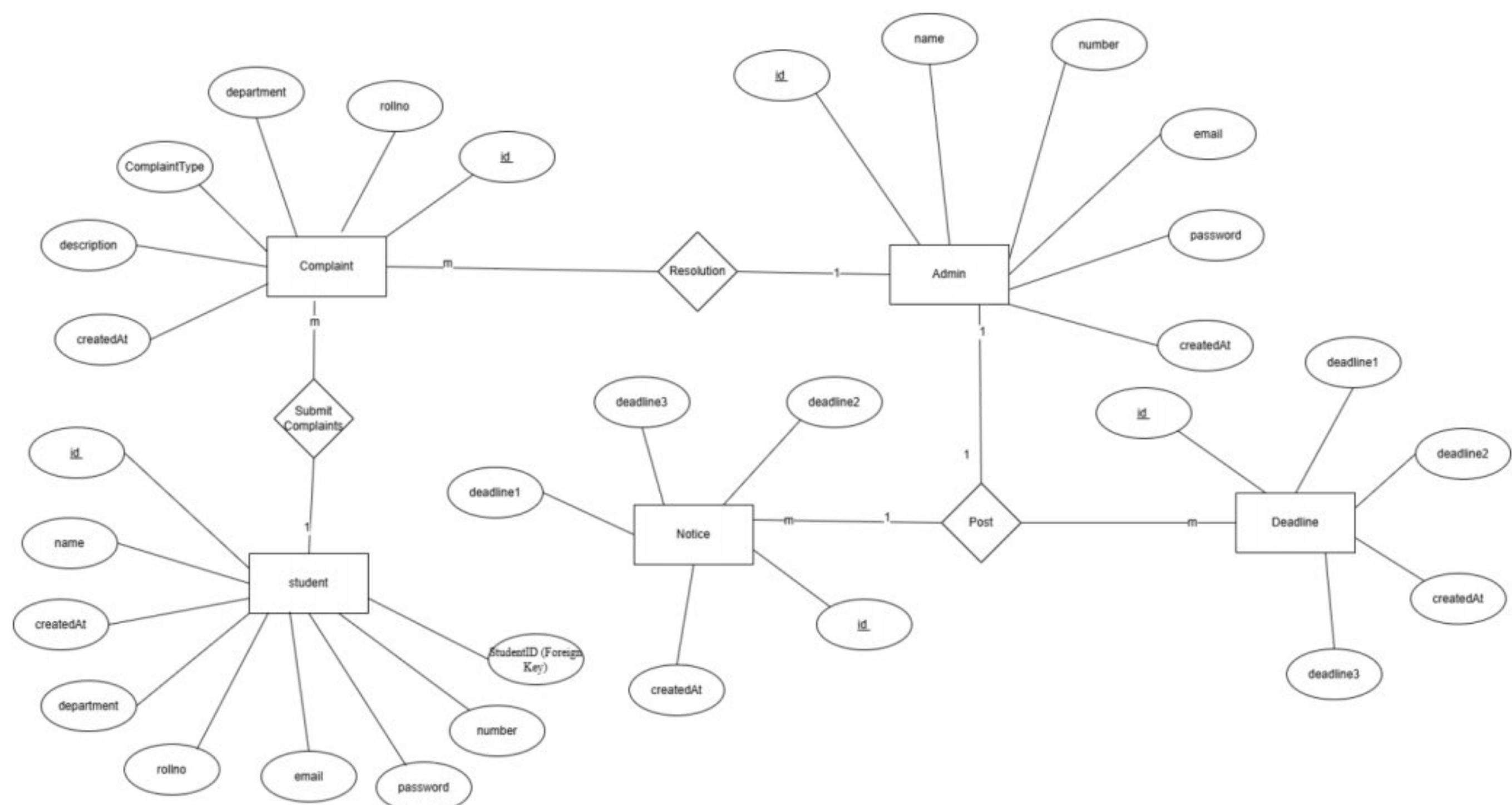
✓ **Scalability:**

The architecture should support the addition of more users (students and admins) and future features without affecting performance.

✓ **Performance:**

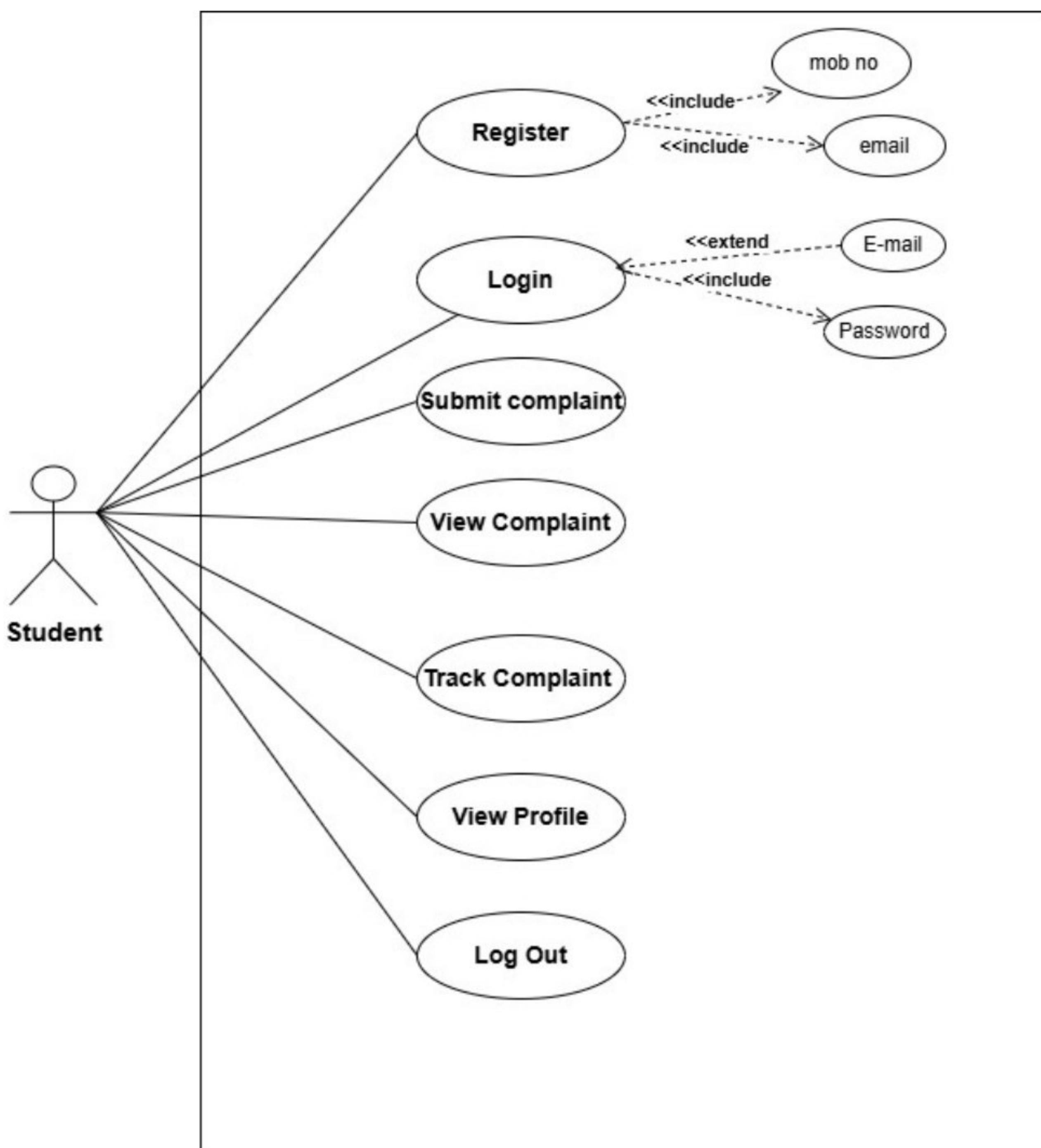
The system should respond quickly to user actions, even under moderate to high load, ensuring a smooth experience.

3.2 ER Diagram

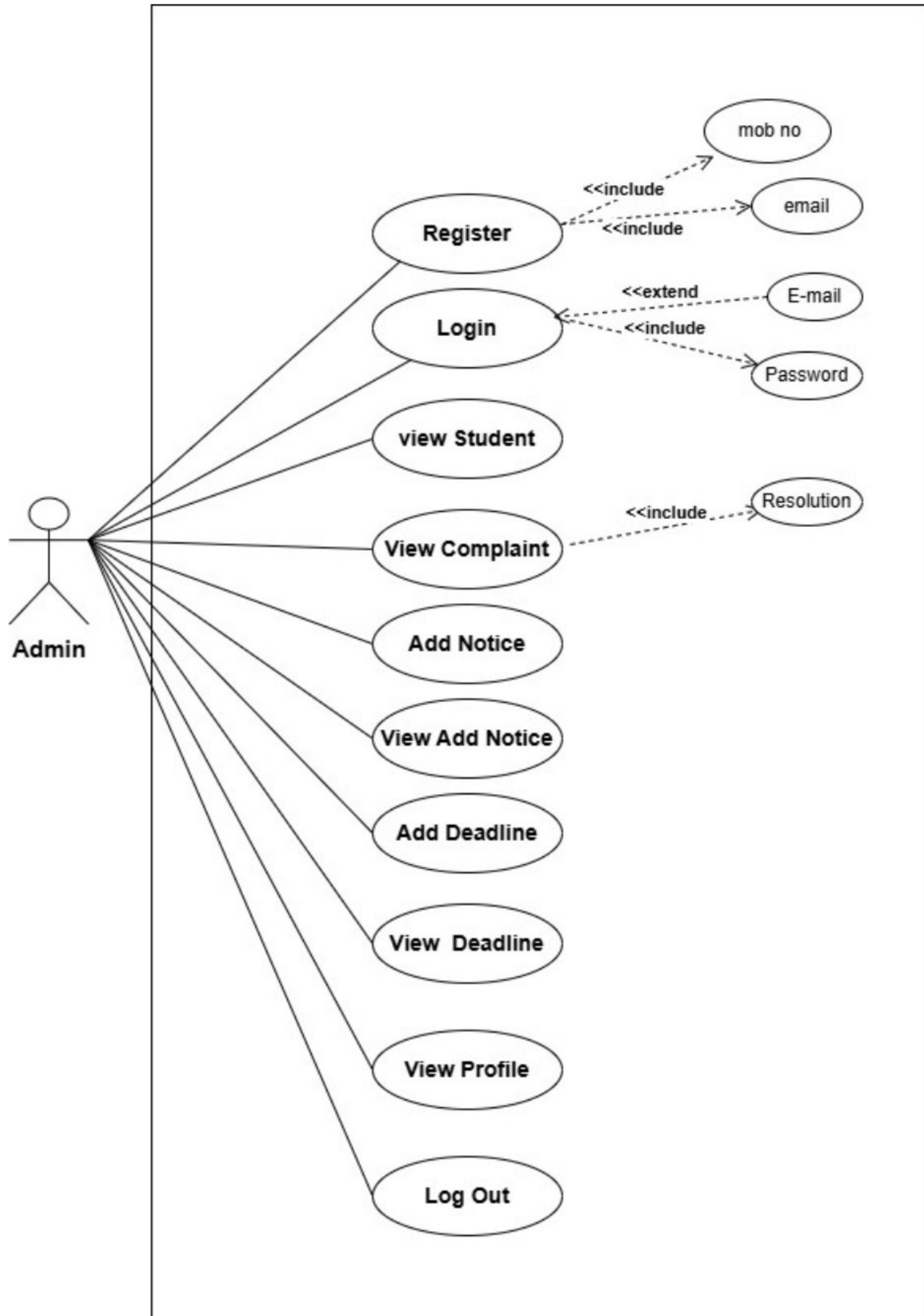


3.3 Use Case Diagrams

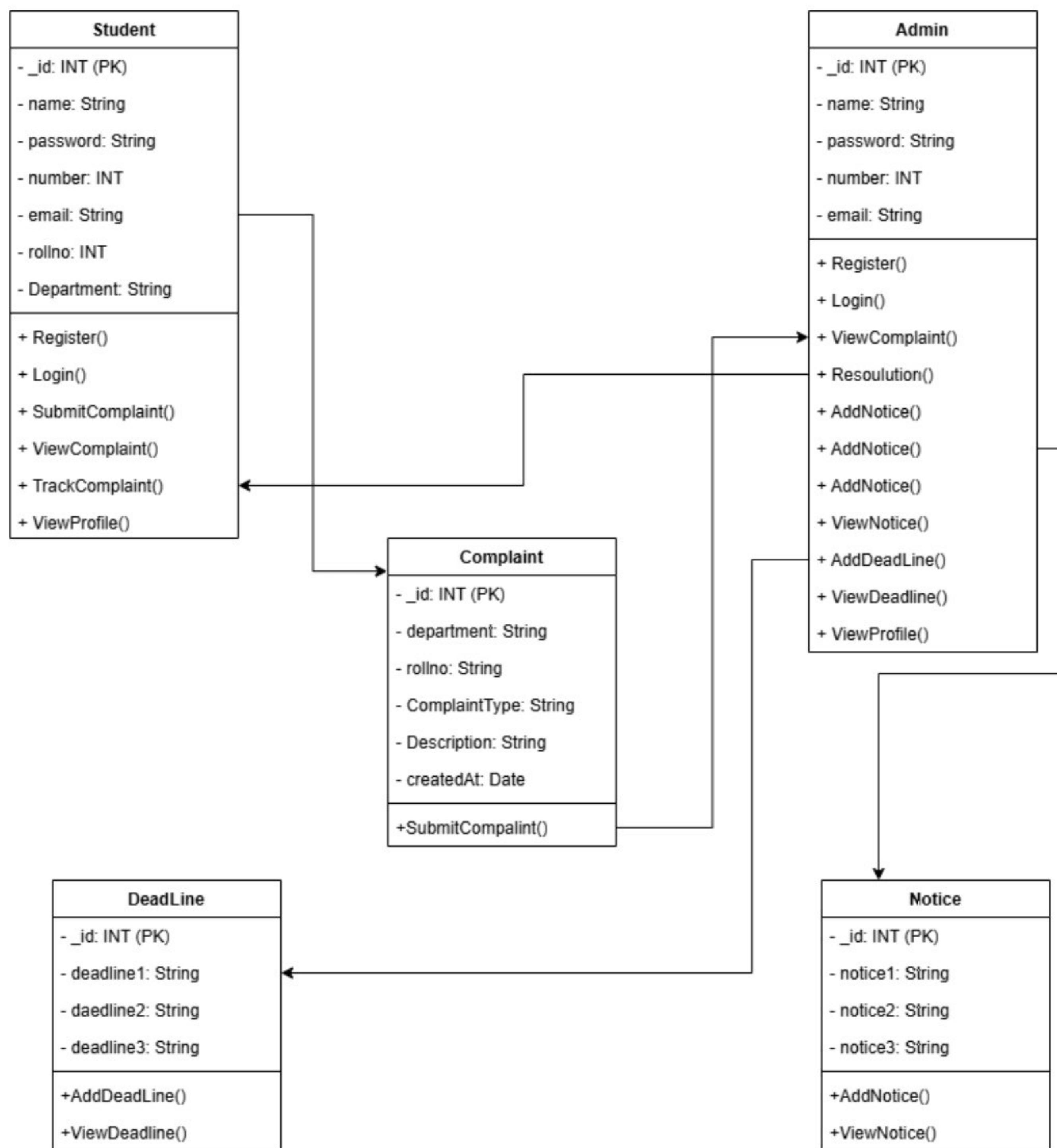
Student



Admin

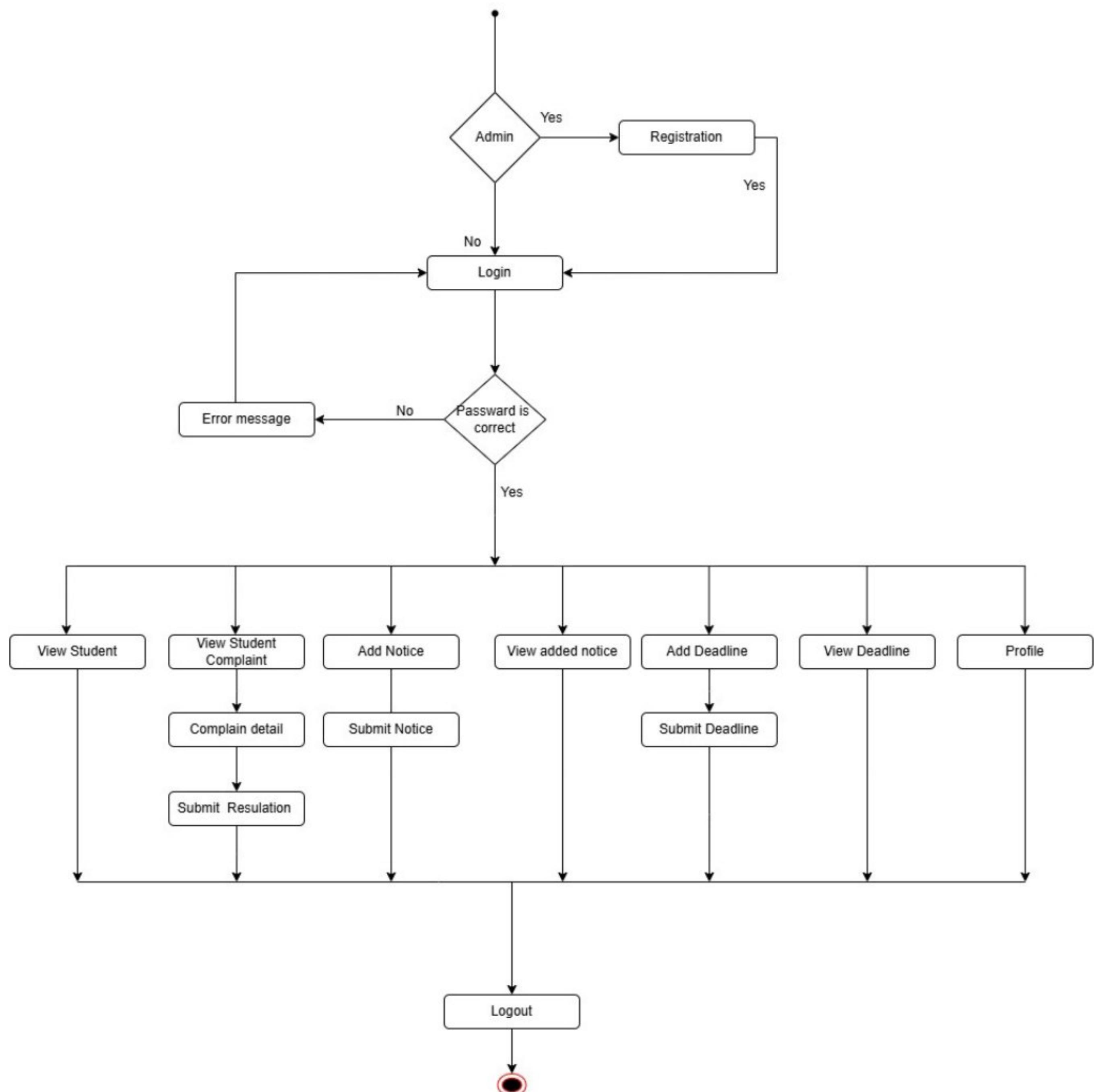


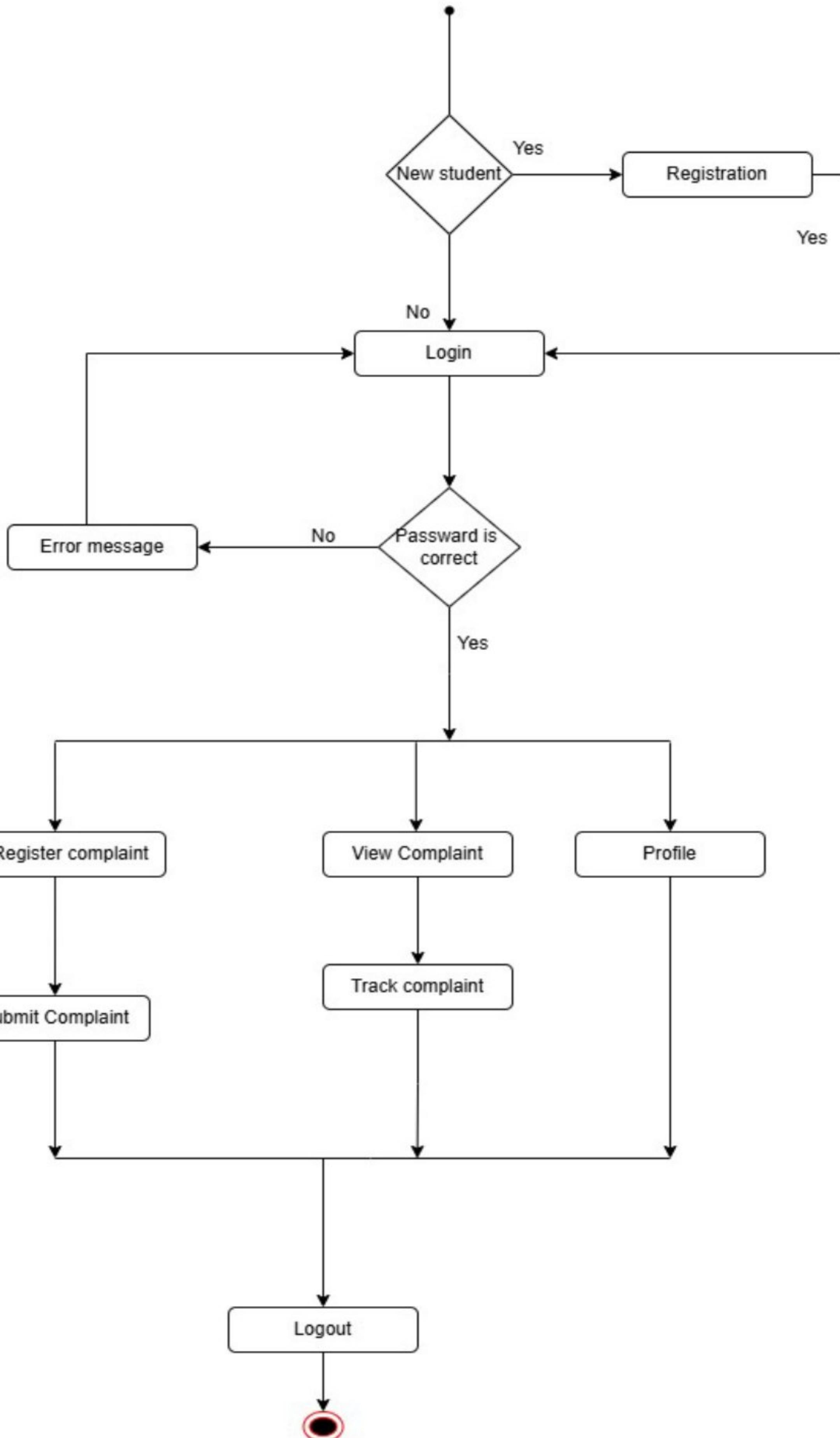
3.4 Class Diagram



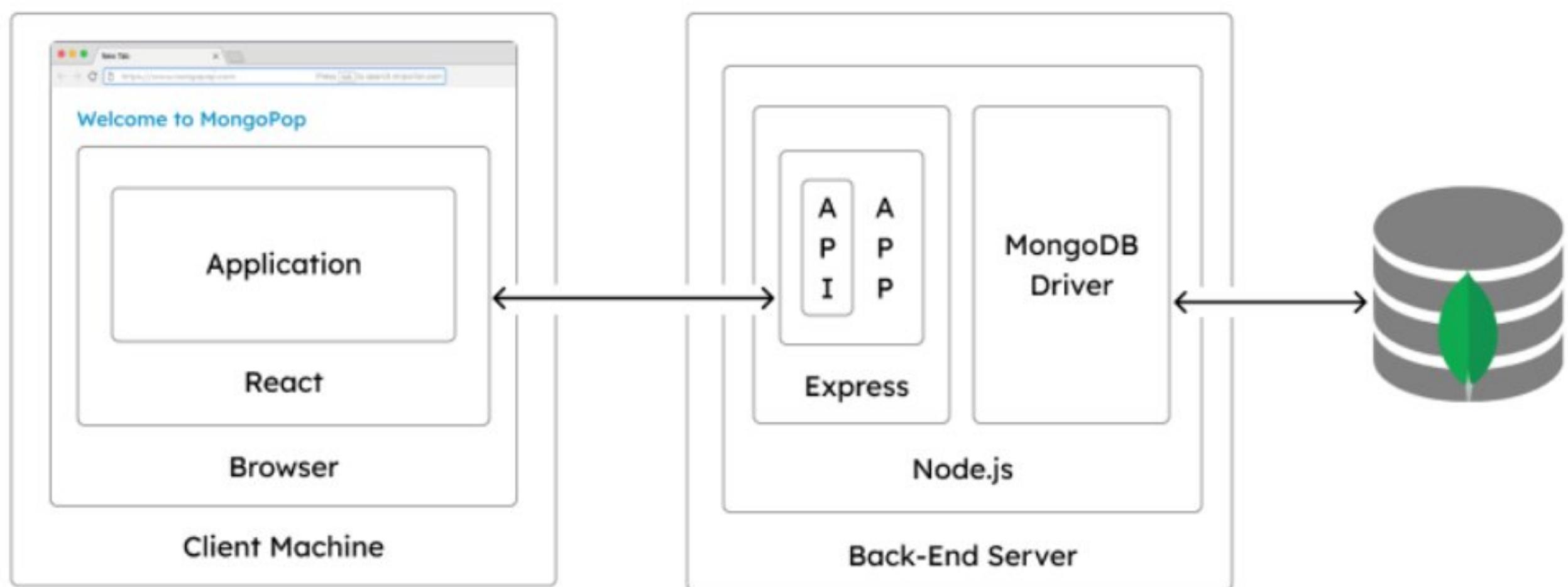
3.5 Activity Diagram

AdminActivity





3.6 Deployment Diagram



3.7 Data Structure:-

Student:-

```
{  
  "_id": {  
    "$oid": "67f7d2b50191e7b6a9845599"  
  },  
  "name": "John Doe",  
  "department": "MCA",  
  "rollno": "CS12345",  
  "email": "john.doe@example.com",  
  "number": "1234567890",  
  "password":  
    "$2b$10$sHXi2I66Bqm8GXn0wbYRSuyZTT8yZiqNIW4p.SNdZIN4ogTyOYZLm  
  ",  
  "createdAt": {  
    "$date": "2025-04-10T14:16:21.913Z"  
  },  
  "__v": 0  
},
```

Complaint:-

```
{  
  "_id": {  
    "$oid": "67f8201fec8592cb7916016c"  
  },  
  "department": "MCA",  
  "rollno": "24254",  
  "complaint": "The student is facing difficulties in understanding the course material."}
```

```
"ComplaintType": "Academic",
"description": "Assignments are being given without proper explanation in
class.",
"createdAt": {
"$date": "2025-04-10T19:46:39.883Z"
},
"__v": 0
},
```

Deadline Model:-

```
[{
"_id": {
"$oid": "67f814d1257cf01a8fce186b"
},
"deadline1": "Infosys Online Test",
"deadline2": " Deadline: April 3, 2025",
"deadline3": "Campus drive scheduled on March 20, 2025. Register via TPO.",
"createdAt": {
"$date": "2025-04-10T18:58:25.497Z"
},
"__v": 0
}]
```

Notice Model:-

```
[{
"_id": {
"$oid": "67f81369257cf01a8fce1861"
},
"notice1": "Internal Exam Schedule Announced - April",
"notice2": "Mid-sem exams begin on April 15th",
}]]
```

```
"notice3": "Check the timetable on the student whatsapp group",
"createdAt": {
  "$date": "2025-04-10T18:52:25.493Z"
},
"__v": 0
}]
```

3.8 Sample Screen

Home page

WELCOME TO DYPIMED PUNE

Grievance Portal



Student Portal

Access your academic records, submit grievances, and track their status

[Login](#)[Register](#)

Admin Portal

Manage Grievance, view student grievances, and provide resolutions

[Login](#)[Register](#)

Latest Notices

Internal Exam Schedule Announced - April

on April 15th

Check the timetable on the student whatsapp group



Important Deadlines

Infosys Online Test

Deadline: April 3, 2025

Campus drive scheduled on March 20, 2025. Register via TPO.



Contact Information

Emergency: +1 (555) 123-4567

Admin Office: +1 (555) 987-6543

Email: dypmied@dypatil.edu



FAQ Section

How to Submit Grievance?

Track Resolution Status

Document Requirements

Key Features



Online Grievance Submission

Submit your concerns easily through our streamlined portal.



Track Grievance Status

Monitor the progress of your grievances with real-time updates.



Resolution Timeline

Get estimates and notifications at each stage of resolution.

Contact Us

123 College, Pune Varale, Talegaon
9881884005
dyp@dypimed.edu

Quick Links

About Us
Academics
Admissions
Campus Life

Follow Us

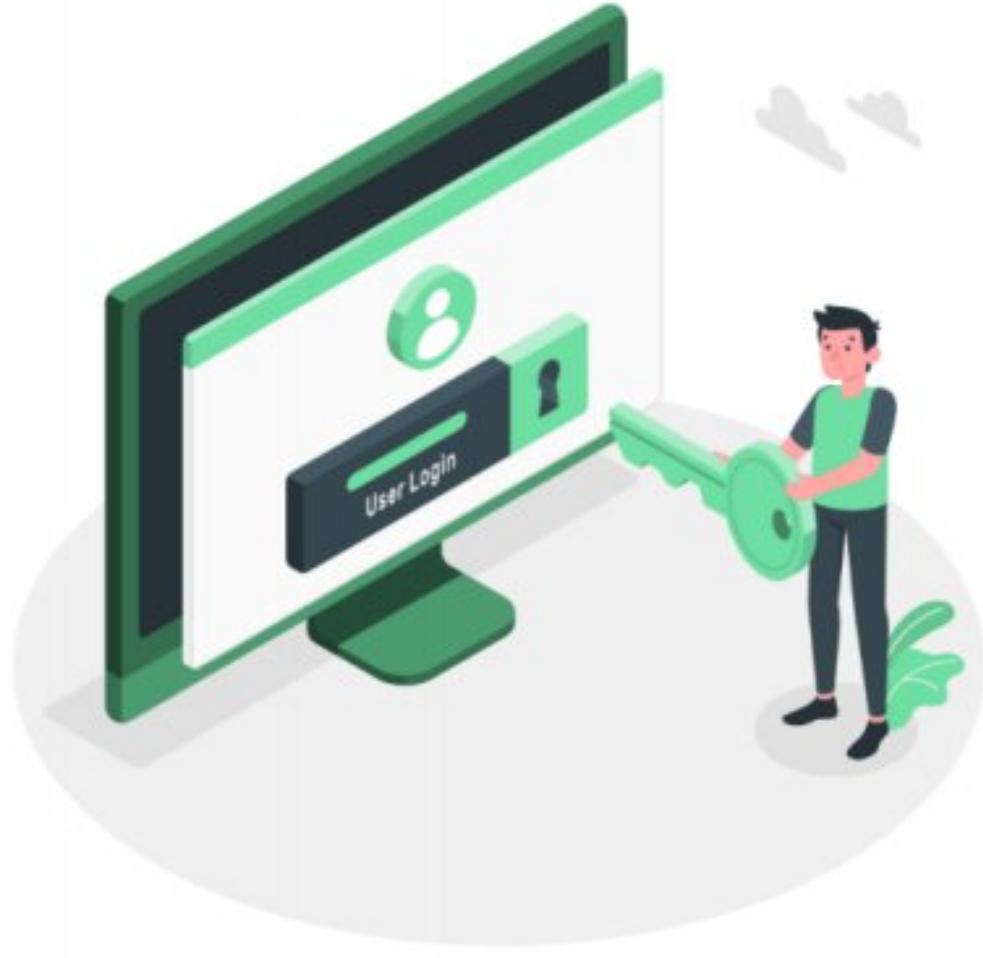
[f](#) [in](#) [i](#) [t](#)

© 2025 DYPIMED College. All rights reserved | Privacy Policy | Terms of Service

Student Login and Register page

Sign In

Hello Student! Welcome Back



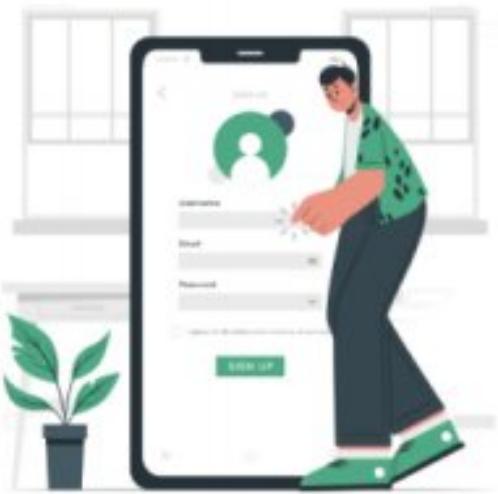
Your Email

Password

Sign In

Don't have an Account?
[Register Here](#)

Create an Account



Name

Department

Roll Number

Email

Contact Number

Password

Sign Up

Already have an account? [Login here](#)

Student Dashboard:-

complaint form

Student Dashboard

- [Register Complaint](#)
- [View Complaints](#)
- [Track Status](#)
- [Your Profile](#)

Student Complaint Form

Department

Roll no

Complaint Type

Description of Complaint

Submitting...

View Complaints

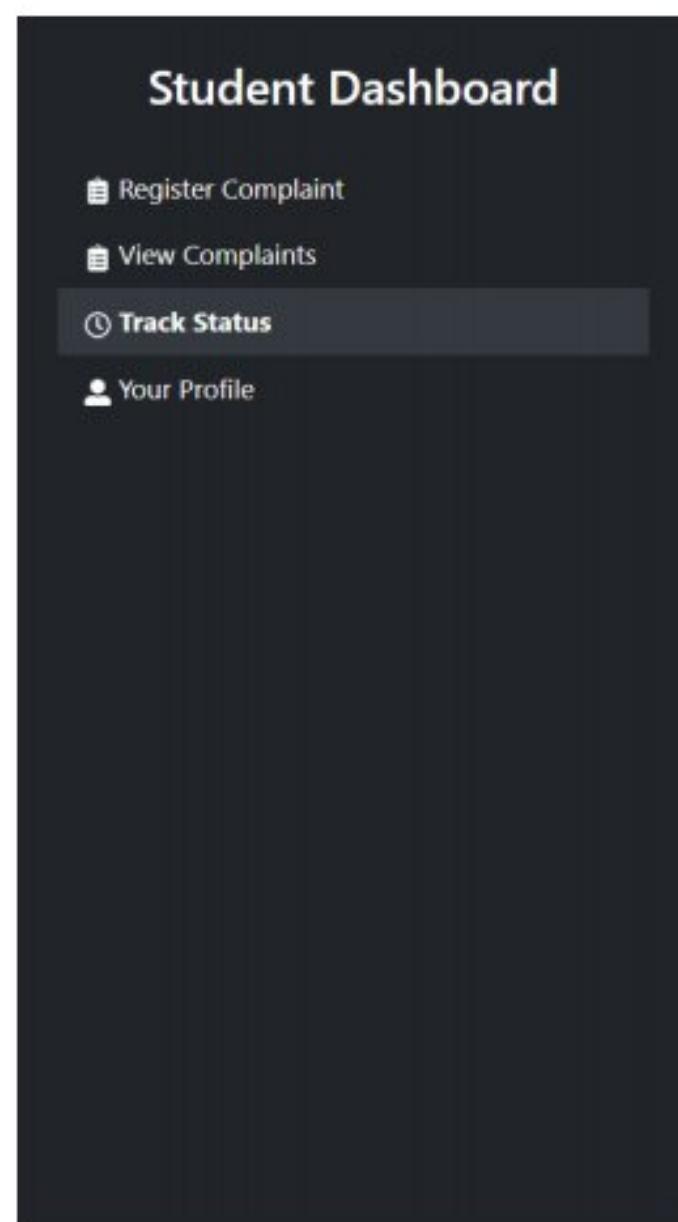
Student Dashboard

- [Register Complaint](#)
- [View Complaints](#)
- [Track Status](#)
- [Your Profile](#)

Student Complaints

Department	Roll no	Complaint Type	Description
MCA	24254	Facilities	Drinking water cooler on 2nd floor is not working.
MCA	24254	Academic	Assignments are being given without proper explanation in class.
MCA	24254	Other	Cafeteria prices have increased without any notice to students.
MCA	24254	Facilities	Fans in Room 204 are making loud noise and need maintenance.
MCA	24254	Academic	Lab sessions are overcrowded and lack proper equipment.

Track Complaint

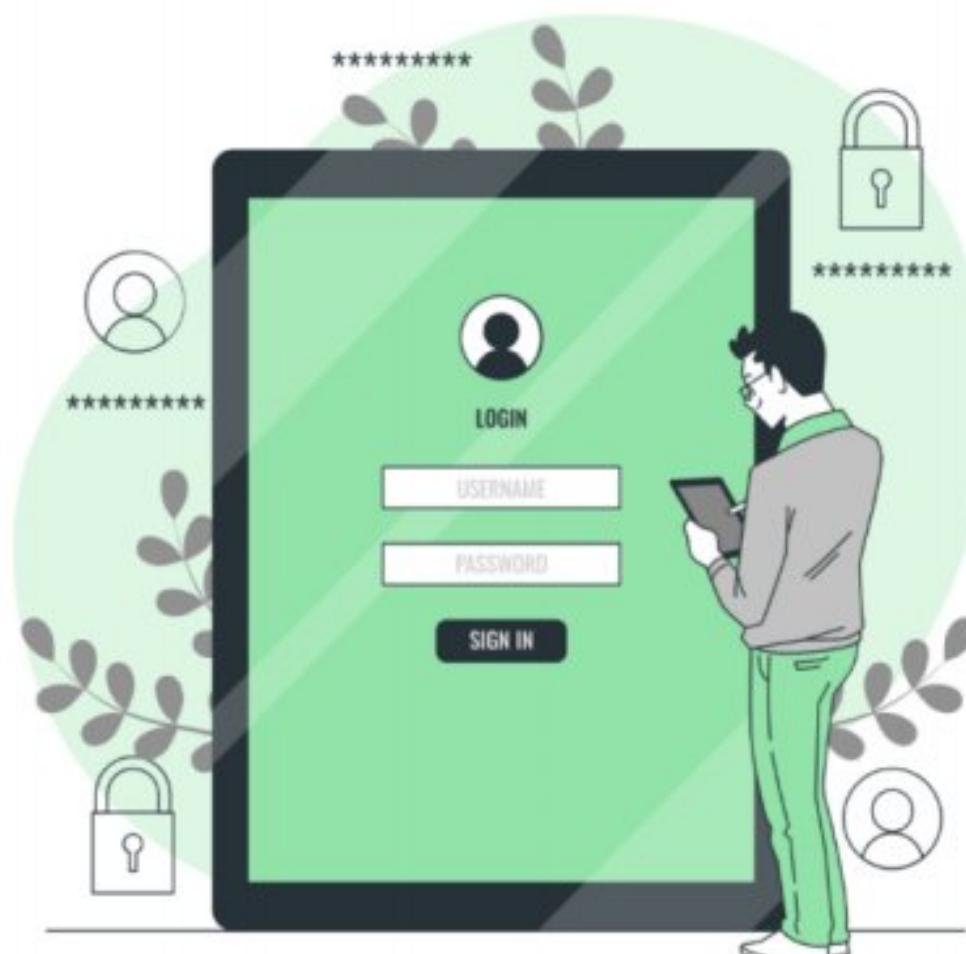


Student Complaints

Department	Roll no	Complaint Type	Actions
MCA	24254	Facilities	Pending
MCA	24254	Academic	Pending
MCA	24254	Other	Pending
MCA	24254	Facilities	Pending
MCA	24254	Academic	Pending

Admin Login

Sign In



Hello Admin
Welcome Back

Your Email

Your Valid Email

Password

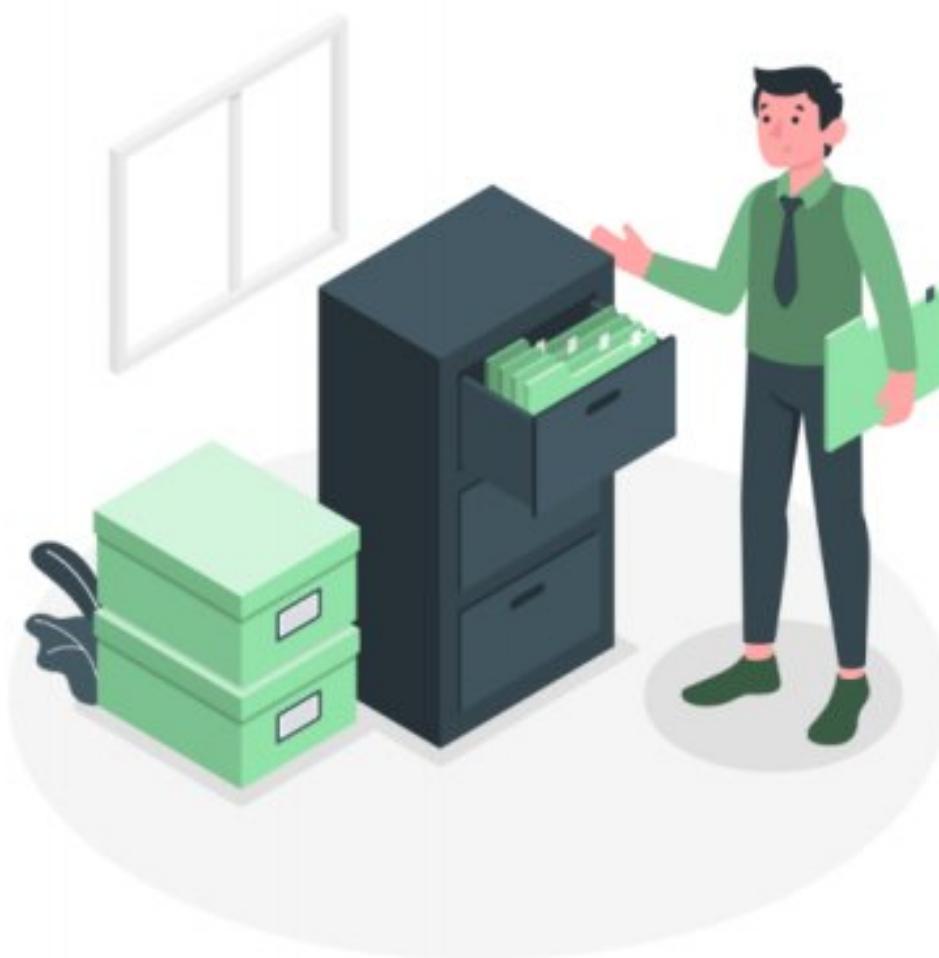
Create Strong Password

Sign In

Don't have an Account

[Register Here](#)

Admin Register



Sign Up

Your Name
admin

Your Email
admin@123gmail.com

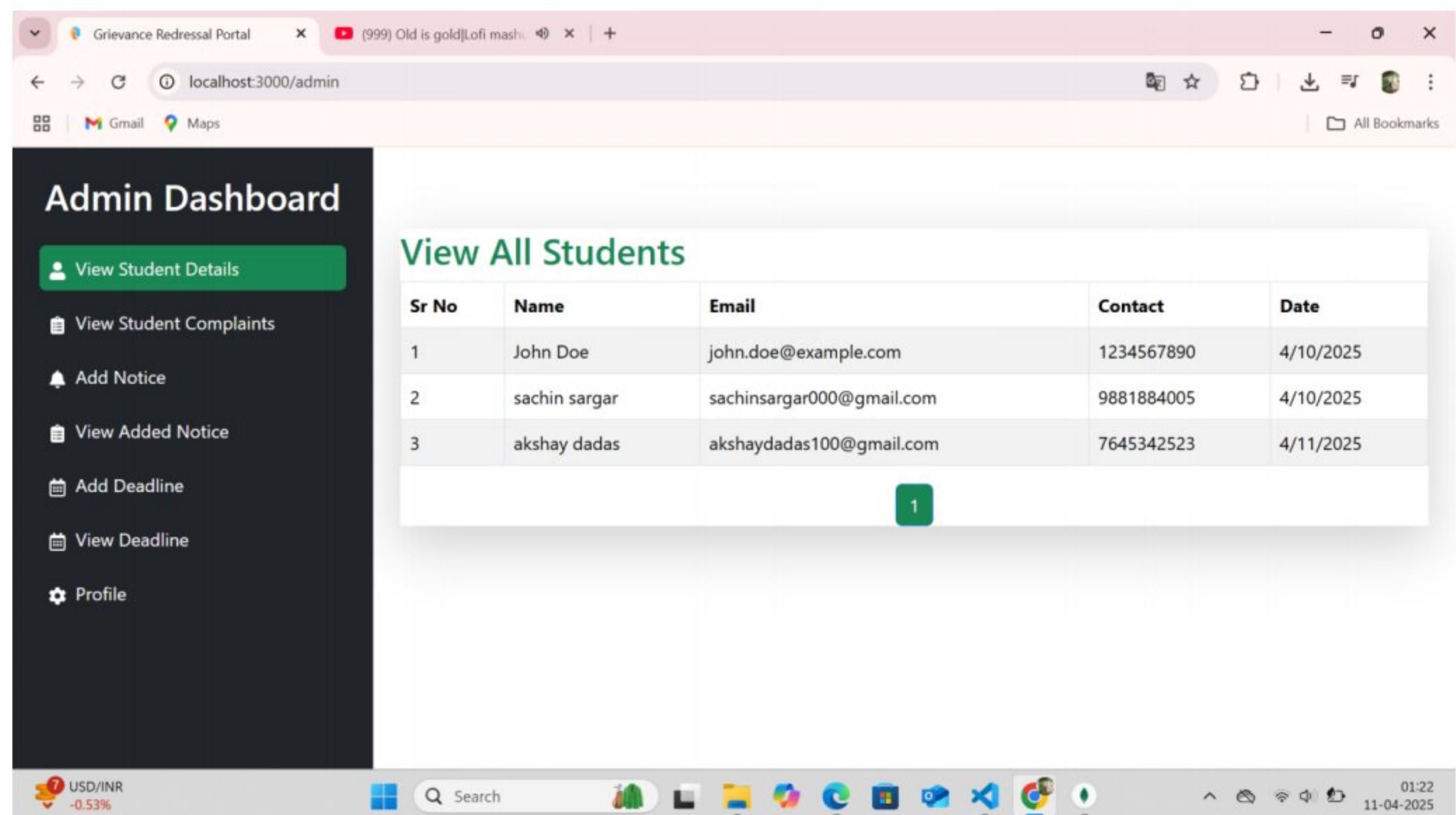
Your Contact Number
8996664532

Password
.....

[Sign Up](#)

Already have an Account
[Login Here](#)

View Student Details



Grievance Redressal Portal

localhost:3000/admin

Admin Dashboard

- [View Student Details](#) (highlighted)
- [View Student Complaints](#)
- [Add Notice](#)
- [View Added Notice](#)
- [Add Deadline](#)
- [View Deadline](#)
- [Profile](#)

View All Students

Sr No	Name	Email	Contact	Date
1	John Doe	john.doe@example.com	1234567890	4/10/2025
2	sachin sargar	sachinsargar000@gmail.com	9881884005	4/10/2025
3	akshay dadas	akshaydadas100@gmail.com	7645342523	4/11/2025

1

USD/INR -0.53% 01:22 11-04-2025

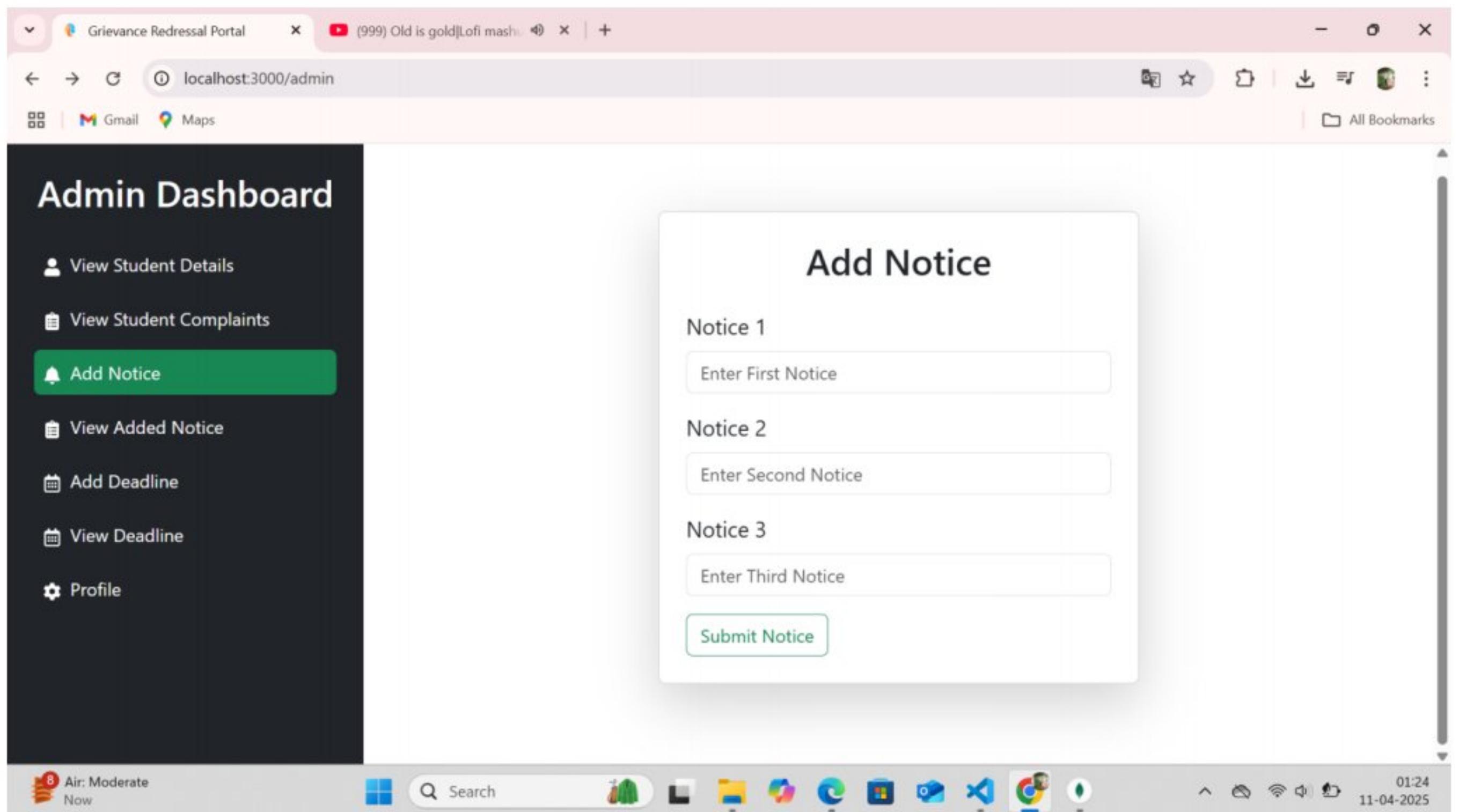
View Student complaint

The screenshot shows a web browser window titled "Grievance Redressal Portal" at "localhost:3000/admin". The left sidebar is titled "Admin Dashboard" and includes links for "View Student Details", "View Student Complaints" (which is highlighted in green), "Add Notice", "View Added Notice", "Add Deadline", "View Deadline", and "Profile". The main content area is titled "Student Complaints" and displays a table of complaints. The table has columns for "Department", "Roll no", "Complaint Type", and "Actions". There are five rows of data:

Department	Roll no	Complaint Type	Actions
MCA	24254	Facilities	Resulation
MCA	24254	Academic	Resulation
MCA	24254	Other	Resulation
MCA	24254	Facilities	Resulation
MCA	24254	Academic	Resulation

The status bar at the bottom shows "localhost:3000/Response/67f8201fec8592cb7916016c" and system icons.

Add Notice



View Notice

The screenshot shows the Admin Dashboard with the 'View Added Notice' link highlighted in green. The main area is titled 'View Notices' and displays a table with four columns: 'SR NO', 'Notice 1', 'Notice 2', and 'Notice 3'. The table has one row with the following data:

SR NO	Notice 1	Notice 2	Notice 3	Date
1	Internal Exam Schedule Announced - April	Mid-sem exams begin on April 15th	Check the timetable on the student whatsapp group	4/11/2025

Add Deadline

The screenshot shows the Admin Dashboard with a dark sidebar on the left containing various administrative tasks. The main area is titled "Important Deadlines". It features three input fields labeled "Deadline 1", "Deadline 2", and "Deadline 3", each with a placeholder text ("Enter First Deadline", "Enter Second Deadline", "Enter Third Deadline") and a "Submit Deadline" button below them.

View Deadline

The screenshot shows the Admin Dashboard with a dark sidebar on the left. The main area is titled "View Deadlines" and displays a table with one row of data. The table has columns for "SR NO", "Important Deadline 1", "Important Deadline 2", "Important Deadline 3", and "Date". The data row shows SR NO 1, Important Deadline 1 as "Infosys Online Test", Important Deadline 2 as "Deadline: April 3, 2025", Important Deadline 3 as "Campus drive scheduled on March 20, 2025. Register via TPO.", and Date as "4/11/2025".

SR NO	Important Deadline 1	Important Deadline 2	Important Deadline 3	Date
1	Infosys Online Test	Deadline: April 3, 2025	Campus drive scheduled on March 20, 2025. Register via TPO.	4/11/2025

4: Coding

4.1 Code Snippets

Backend Controller to Submit Complaint:

```
import StudentComplaint from "../../Model/student/SMComplaint.js";
import students from "../../Model/student/StudentRegisterModel.js";

//POST endpoint to create a new complaint for a specific student
export const createComplaint = async (req, res) => {
  try {
    const { studentId, department, rollNo, ComplaintType, description } =
      req.body;

    // Find the user by rollNo and email
    const user = await students.findOne({ rollNo });

    if (!user) {
      return res.status(404).json({ message: "User not found" });
    }

    const newComplaint = new StudentComplaint({
      studentId,
      department,
      rollNo,
      ComplaintType,
      description,
      timestamp: Date.now(),
    });

    await newComplaint.save();
    res.status(201).json(newComplaint);
  } catch (error) {
    console.error(error);
    res.status(500).json({ message: "Internal Server Error" });
  }
};
```

```

}

//Create the complaint with user reference
const complaint = new StudentComplaint({
  user: user._id,
  studentId,
  department,
  rollno,
  ComplaintType,
  description,
});

await complaint.save();

res.status(201).json({ message: 'Complaint submitted', complaint });
} catch (err) {
  console.error(err);
  res.status(500).json({ message: 'Server error' });
}
};

//GET endpoint to retrieve all complaints
export const getComplaints = async (req, res) => {
  try {
    const complaints = await StudentComplaint.find();

    if (complaints) {
      res.status(200).json(complaints);
    } else {
      res.status(500).json({ msg: "No Complaints Found" });
    }
  } catch (err) {
    console.error("Error fetching complaints:", err); // Log the error for better debugging
    res.status(500).json({ msg: 'Failed to retrieve complaints.' });
  }
};

export const getComplaint = async (req, res) => {
  try {
    const complaint = await StudentComplaint.findById(req.params.id);

    if (complaint) {

```

```

res
  .status(200)
  .json({msg: "complaint accessed successfully", complaint});
} else{
  res.status(404).json({msg: "complaint not found"});
}
} catch(err){
  console.error(err);
  res.status(500).json({message: "Failed to retrieve complaint."});
}
};

export const getStdComplaint = async (req, res) => {
  try{
    const complaint = await StudentComplaint.findById(req.params.id);

    if(complaint){
      res
        .status(200)
        .json({msg: "complaint accessed successfully", complaint});
    } else{
      res.status(404).json({msg: "complaint not found"});
    }
  } catch(err){
    console.error(err);
    res.status(500).json({message: "Failed to retrieve complaint."});
  }
};

```

Frontend - Submit Complaint Form (React):

```

<form
  onSubmit={handleSubmit}
  className="col-md-6 mx-auto p-4 border rounded shadow-lg"
>
  <h2 className="text-center mb-4">Student Complaint Form</h2>

  <div className="mb-3">

```

```

<label htmlFor="description" className="form-label">
  Department
</label>
<input
  type="text"
  name="department"
  id="department"
  value={formData.department}
  onChange={handleChange}
  placeholder="Your Department"
  rows="4"
  required
  className="form-control"
/>

```

```

<label htmlFor="description" className="form-label">
  Roll no
</label>
<input
  type="text"
  name="rollno"
  id="rollno"
  value={formData.rollno}
  onChange={handleChange}
  placeholder="Your Roll No"
  rows="4"
  required
  className="form-control"
/>
</div>
/* Complaint Type Dropdown */
<div className="mb-3">
<label htmlFor="ComplaintType" className="form-label">
  Complaint Type
</label>
<select
  name="ComplaintType"
  id="ComplaintType"
  value={formData.ComplaintType}
  onChange={handleChange}
  required
  className="form-select"
>
<option value="Academic">Academic</option>
<option value="Facilities">Facilities</option>
<option value="Other">Other</option>

```

```

        </select>
    </div>

    /* Complaint Description */
    <div className="mb-3">
        <label htmlFor="description" className="form-label">
            Description of Complaint
        </label>
        <textarea
            name="description"
            id="description"
            value={formData.description}
            onChange={handleChange}
            placeholder="Describe your complaint"
            rows="4"
            required
            className="form-control"
        />
    </div>

    /* Error Message */
    {error && <div className="alert alert-danger">{error}</div>}

    /* Success Message */
    {successMessage && (
        <div className="alert alert-success">{successMessage}</div>
    )}

    /* SubmitButton */
    <button
        type="submit"
        disabled={isLoading}
        className="btn btn-outline-success w-100"
    >
        {isLoading ? "Submitting..." : "Submit Complaint"}
    </button>
</form>

```

5. Testing

5.1 Test Cases (Detailed)

Test ID	Description	Input	Expected Output	Status
TC0 01	Student Login	Email, Password	Login Success or Error Message	Pas s
TC0 02	Submit Complaint	Valid Title, Description	Complain t Successf ully Stored	Pas s
TC0 03	Submit Complaint (Missing Fields)	Missing Title or Description	Error Message Promptin g Required Fields	Pas s
TC0 04	View Complaint Status	Student Logged In	List of Student's Complaints with	Pas s

Test ID	Description	Input	Expected Output	Status
			Status	
TC0 05	Admin Login	Admin Email, Password	Admin Dashboard Access or Error Message	Pas s
TC0 06	Admin View Complaints	Admin Logged In	Display All Complaints from All Students	Pas s
TC0 07	Update Complaint Status	Complaint ID, New Status	Status Updated in Complaint Record	Pas s
TC0 08	Add Resolution to Complaint	Complaint ID, Resolution Text	Resolution Stored & Linked to Complaint	Pas s
TC0	Post	Title,	General	Pas

Test ID	Description	Input	Expected Output	Status
09	General Notice	Content, Type = General	Notice Displayed to Students	s
TC0 10	Post Important Notice	Title, Content, Type = Important	Important Notice Marked and Displayed	Pass
TC0 11	Post Deadline	Title, Description, DueDate	Deadline Displayed to Students	Pass
TC0 12	View Notices	None	All General and Important Notices Displayed	Pass
TC0 13	View Deadlines	None	List of All Deadlines Posted by Admin	Pass
TC0	Invalid	Wrong	Error	Pass

Test ID	Description	Input	Expected Output	Status
14	Login Attempt	Email or Password	Message: "Invalid Credentials"	s
TC0 15	Duplicate Complaint Submission	Same title + description twice	Optional: Warn or Allow Based on Rule	Pass
TC0 16	Student View Profile	Student Logged In	Display Student Info (Name, Email, etc.)	Pass
TC0 17	Admin View Students	Admin Logged In	List of All Registered Students	Pass
TC0 18	Logout Functionality	Logout Button Clicked	Redirect to Login Page, Session Cleared	Pass

Test ID	Description	Input	Expected Output	Status
TC0 19	Unauthorized Admin Access	Direct URL Access Without Login	Redirect to Admin Login Page / Access Denied	Pas s
TC0 20	Backend API Call (Complaint Submit)	POST /api/complaint with valid payload	Status 200 with confirmation message	Pas s

6: Limitations of Proposed System

- No support for complaint categorization by department
- No escalation workflow for unresolved complaints
- Limited notification system (no email/SMS)
- No user feedback system after complaint resolution

7: Proposed Enhancements

- Integrate email/SMS notification system
- Add complaint categories and departments
- Implement escalation policies and response time tracking
- Collect user feedback on resolved complaints
- Add analytics dashboard for admins

8: Conclusion

The Grievance Redressal College Portal provides a much-needed solution to the inefficiencies of the traditional paper-based system. It improves transparency, accountability, and communication between students and college authorities. By using modern web technologies and providing real-time updates, this system creates a more reliable environment for grievance management. Future updates can enhance its capabilities and expand its usage across different departments and institutions.

9: Bibliography

- MongoDB Official Docs (<https://mongodb.com>)
- React Documentation (<https://react.dev>)
- Node.js Documentation (<https://nodejs.org>)
- Express.js Guide (<https://expressjs.com>)
- Research Papers on Grievance Systems

10: User Manual

For Students:

1. Visit the portal and register.
2. Log in with your credentials.
3. Click on "Submit Complaint" and fill in the details.
4. Check the status of your complaints in the "Track Complaint" section.
5. View general and important notices on your dashboard.

For Admins:

1. Log in using admin credentials.
2. View list of student users.
3. Open the complaint dashboard to see new complaints.

4. Click on a complaint to read details and update status.
5. Navigate to "Add Notice" to post a regular or important notice.
6. All notices will be visible to students on their dashboard.