DESIGN AND IMPLEMENTATION OF A WEB-BASED REPORTING AND CASE-MANAGEMENT SYSTEM FOR SOCIAL INJUSTICE IN A NIGERIAN UNIVERSITY: A STUDY OF THE UNIVERSITY OF BENIN

BY

AISOSA OSAYOMWANBOR
PSC1605647

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF COMPUTING
UNIVERSITY OF BENIN
BENIN CITY
EDO STATE

NOVEMBER 2025

DESIGN AND IMPLEMENTATION OF A WEB-BASED REPORTING AND CASE-MANAGEMENT SYSTEM FOR SOCIAL INJUSTICE IN A NIGERIAN UNIVERSITY: A STUDY OF THE UNIVERSITY OF BENIN

BY

AISOSA OSAYOMWANBOR
PSC1605647

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTING, UNIVERSITY OF BENIN, BENIN CITY, EDO STATE, NIGERIA,

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE

NOVEMBER 2025

CERTIFICATION

This is to certify that this project work was carried out by **AISOSA OSAYOMWANBOR** with Matriculation Number **PSC1605647** under my supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.Sc.) Degree in Computer Science of the University of Benin.

_____ _____

Mr. G.I. EVBOUMWAN **DATE**

Project supervisor

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

_____ _____

Dr. (Mrs.) A.R. USIOBAIFO **DATE**

Head of Department

DEDICATION

This project is dedicated to my family, for their unwavering support and encouragement, and to all individuals currently fighting for social justice, whose courage and resilience inspire change.

ACKNOWLEDGEMENT

First and foremost, I express my profound gratitude to the Almighty God for His infinite grace, wisdom, and protection throughout my academic pursuit.

I am deeply indebted to my project supervisor, Mr.Greg Evboumwan, for his scholarly guidance, constructive criticism, and patience during the course of this research. I also extend my sincere appreciation to the Head of Department, Dr. (Mrs.) Usiobaifo, for her exemplary leadership and the supportive academic environment she has fostered.

My heartfelt thanks go to my family for their unwavering belief in me, especially my parents, Hon. Leo Osayomwanbor Eboigbe and Mrs. Tina Ikpomosa Osayomwanbor; my in-law, Mr. Godwin Akhigbe; my sisters, Mrs. Oyemwen Akhigbe and Ms. Uyiosa Osayomwanbor; and my aunt, Mrs. Charity Ekhator. Your unwavering love and selfless sacrifices have been my greatest source of strength

I also wish to acknowledge the support of my friends Victory "Sage" Erebor, Alex Obarentin, Jeffery Iyoha, Sonia Emon, Offonimeh Akai, Pearl Makavar, Naomi Erewa, Godstime Osagede, Osawie "Reggie" Osagie, and Zino Gbobodo, as well as my WhatsApp community, for their valuable contributions, encouragement, and companionship.

Finally, a special thank you to Prof. Ekuobase G.O. for his enduring support and mentorship throughout my journey in Computer Science.

Table of Contents

TITLE PAGE i

CERTIFICATION ii

APPROVAL iii

DEDICATION iv

ACKNOWLEDGEMENT v

TABLE OF CONTENTS vi

LIST OF TABLES ix

LIST OF FIGURES x

ABSTRACT xii

CHAPTER ONE 1

INTRODUCTION 1

1.1 Background of the Study 1

1.2 Statement of the Problem 2

1.3 Aim and Objectives of the Study 3

1.4 Research Questions 4

1.6 Scope of the Study 4

1.7 Limitations of the Study 5

1.8 Organization of the Study 6

1.9 Definition of Terms 7

CHAPTER TWO 9

LITERATURE REVIEW 9

2.1 Introduction 9

2.2 Conceptual Framework: Social Injustice 10

2.2.1 Forms of Social Injustice in Nigerian Universities 11

2.2.2 Classification of Injustice 12

2.3 Conceptual Framework: Reporting Systems 13

2.3.1 Defining Reporting Systems 13

2.3.2 Typologies of Reporting Systems 14
2.4 Theoretical Frameworks 15
2.5.1 Prevalence and Reporting Barriers in Nigeria 17
2.5.2 Technology Pilots and System Evaluations 18
2.7 Synthesis and Identified Gaps 20
CHAPTER THREE 22
SYSTEM ANALYSIS AND DESIGN METHODOLOGY 22
3.1 Introduction 22
3.2 Materials and Methods 22
3.3 Sampling Method 24
3.5 Study Instruments 24
3.6 Questionnaire Design 24
3.7 Data Distribution and Collection 25
3.8.1 Demographic Data 26
3.8.2 Awareness and Experience of social injustice 30
3.8.1 Reporting Practices and Challenges 34
3.8.3 System Usability and Feature Evaluation 42
3.11 System Design 48
3.12 System Design Tools 49
3.12.1 System Design Tool: UML 49
3.12.2 UML Use Case Diagram 50
3.12.3 Entity Relationship Diagram 54
CHAPTER FOUR 57
SYSTEM IMPLEMENTATION 57
4.1 Software Implementation Tools 57
4.2.1 Purpose of System Testing 60
4.2.2 Testing Approach 60
4.2.3 Modules Tested 60
4.3 Screenshots of Running Systen 63
CHAPTER FIVE 82
SUMMARY, CONCLUSION AND RECOMMENDATION 82
5.1 Summary of Findings 82

5.2 Conclusion 83
5.3 Limitations of the Study 83
5.4 Recommendations 84
5.5 Summary Statement 85
REFERENCE.....86
APPENDIX.....89
SOURCE CODE.....89

LIST OF TABLES

Tables	Pages
Table 3.1: Age of Participants	27
Table 3.2: Gender of Participants	28
Table 3.3: Level of Study	29
Table 3.4: Residence	30
Table 3.5: Number of those who have experienced social injustice	31
Table 3.6: Types of injustice experienced	32
Table 3.7: Frequency of incidents	33
Table 3.8: Tables showing how students rate the university's current effort at handling issues of social injustice	34
Table 3.9: Table showing the Frequency of reporting	35
Table 3.10: Table showing reasons for not reporting	36
Table 3.11: Table showing the knowledge of an existing reporting channel	37
Table 3.12: Table showing willingness to use an anonymous digital reporting platform	39
Table 3.13: Table showing key features most important of a reporting system	40
Table 3.14: Table showing change in trust level if the university decided to use a transparent online reporting platform	41
Table 3.15: Table showing Features participant considered most important for the web-based system	42
Table 3.16 Table showing User interface design preference	44
Table 3.17 Use Case Notations and Descriptions	51

LIST OF FIGURES

Figures	Pages
Figure 3.1: Chart showing The Age of Participants	28
Figure 3.2: Chart Showing Gender Distribution	29
Figure 3.3: Chart Distribution of Level of Study	30
Figure 3.4: Chart showing residence distribution	31
Figure 3.5: Chart showing Distribution of social injustice experience	32
Figure 3.6: Chart showing Common types of injustice experienced	33
Figure 3.7: Chart showing Frequency of incident	34
Figure 3.8: Chart showing how students rate the current effort at handling issues of social injustice	35
Figure 3.9: Chart showing frequency of reporting injustice	36
Figure 3.10: Chart showing reasons not reporting	37
Figure 3.11: Chart showing of existing reporting channel	38
Figure 3.12: Chart showing willingness to use an anonymous digital reporting platform	39
Figure 3.13: Chart showing features most important for a reporting system	40
Figure 3.14: Chart showing the change in trust if the university uses an online reporting system	41
Figure 3.15 Chart showing features desired in the proposed web-based reporting system	41
Figure 3.16 Chart showing user interface design preference	44
Figure 3.17 Proposed System Architecture Diagram	48
Figure 3.18 Case diagram	53
Figure 3.19: Entity Relationship Diagram of the Speak Up Platform	54
Figure 4.1 Homepage	63
Figure 4.2 Anonymous Reporting page.	65
Figure 4.3 Successful Anonymous report	66
Figure 4.4 Authenticated User Registration page.1	67
Figure 4.5 Authenticated user Registration page 2	68

Figure 4.6 Authenticated User Login page 69
Figure 4.6 Authenticated Reporting page 70
Figure 4.7 Successful Authenticated Report 71
Figure 4.8 User Dashboard 72
Figure:4.9 User Dashboard Report Status Tracking 73
Figure 4.10 Admin dashboard 74
Figure 4.12 Admin dashboard: Case Management{B] 75
Figure 4.13 Case Management: Case Details 76
Figure 4.14 Case Details showing Admin internal notes. 77
Figure 4.15 Case Details showing Admin Public notes. 78
Figure 4.16 Analytic dashboard 79
Figure 4.17 Public Displayed Resolved Case 80
Figure 4.18: Backend Admin dashboard 81

ABSTRACT

Incidents of social injustice, including discrimination, harassment, and unfair treatment, are common in Nigerian tertiary institutions but are often underreported due to fear of retaliation, lack of confidentiality, or inadequate reporting mechanisms. This project presents the development of the SpeakUp App, a web-based platform designed to enable students to report social injustices securely, anonymously, and efficiently. Data for this study was collected via a structured survey administered to students, providing insights into the challenges they face in reporting incidents and their expectations from a reporting system. The system incorporates features for submitting incident reports, attaching evidence, tracking report status, and providing administrative oversight, ensuring complaints are managed effectively while protecting user privacy. Developed using Django for the backend and ReactJS for the frontend, the application leverages Django's security framework and a responsive, user-friendly interface. Survey findings informed the design, and system testing confirmed usability and reliability. The results indicate that the SpeakUp App can significantly improve the reporting and management of social injustice incidents, empowering students and promoting fairness within academic environments.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Universities serve as critical sites for both intellectual advancement and social development, functioning as microcosms of the broader society. Consequently, they reflect the structural and cultural challenges present in wider social contexts, including various forms of social injustice. In Nigerian higher education institutions, such injustices manifest as harassment, discrimination, bullying, and inequitable administrative or academic practices (Dibie & Robert, 2020). These challenges not only undermine the principle of education as an equalizing force but also impede the overall development and well-being of students and staff.

Empirical studies within Nigerian tertiary institutions have highlighted the persistence of inequitable treatment. For example, research conducted in Bayelsa State revealed that students frequently encountered inadequate learning facilities, arbitrary financial charges, and uneven access to institutional resources, all of which constitute forms of social injustice (Obi & Ziprebo, 2024). Such conditions compromise fairness and equity within educational systems, limiting students' opportunities to thrive academically and socially.

Traditional mechanisms for reporting grievances in many Nigerian universities are largely manual, requiring physical visits to administrative offices or submission of written petitions. These approaches are often criticized for being inefficient, lacking confidentiality, and offering no clear mechanism for tracking the progress of reported cases. As a result, underreporting is pervasive, with victims reluctant to come forward due to fear of retaliation, stigmatization, or disbelief (Baba, 2024). This "culture of silence" allows systemic issues to persist unaddressed, adversely affecting the psychological welfare of students and staff alike.

Digital interventions present a viable solution to these challenges. Mobile and web-based platforms have demonstrated the potential to enhance accessibility, confidentiality, and responsiveness in reporting incidents of social injustice, particularly in contexts where traditional reporting mechanisms are inadequate (Edosomwan, 2024). By providing anonymous submission channels and centralized case management, such systems can empower victims, improve administrative oversight, and facilitate timely responses to complaints.

In light of these considerations, there is a compelling need for the University of Benin (UNIBEN) to adopt a web-based system tailored to its specific infrastructural and operational context. By leveraging digital technologies, UNIBEN can strengthen its capacity to manage and respond to social injustice effectively, ensuring a safer, more equitable, and inclusive academic environment.

1.2 Statement of the Problem

Despite the recognized prevalence of social injustice in Nigerian tertiary institutions, the University of Benin (UNIBEN) currently lacks a centralized, confidential, and technologically-enabled mechanism for reporting and managing such incidents. Traditional reporting methods, which often rely on physical visits to administrative offices, written petitions, or verbal communication with faculty members, present several critical limitations:

Underreporting of Incidents: Many students are hesitant to report harassment, discrimination, or bullying due to fear of retaliation, stigmatization, or lack of assurance regarding confidentiality. The absence of secure, anonymous reporting channels means that numerous cases remain unrecorded and unresolved (Baba, 2024).

Inefficient and Non-Transparent Processes: Manual reporting procedures are often slow and inconsistent. Reports can be delayed, misplaced, or handled irregularly, leaving victims uninformed about the progress or resolution of their complaints (Obi & Ziprebo, 2024). This

lack of transparency diminishes trust in the system and discourages students from engaging with existing institutional channels.

Lack of Data-Driven Insights: Without a centralized repository, administrators cannot effectively aggregate, analyze, or interpret incident reports. This limits the ability to identify recurring patterns, track repeat offenders, or pinpoint high-risk locations and faculties. The absence of structured data hinders proactive policy-making and reduces the university's capacity to implement preventive measures (Dibie & Robert, 2020).

These combined limitations create an environment where social injustices can persist unchallenged. Victims receive inadequate support, while the administration is deprived of the tools necessary to enforce accountability and foster a culture of fairness. This study therefore addresses the urgent need for a web-based reporting and management system that enhances confidentiality, operational efficiency, and evidence-based decision-making at UNIBEN.

1.3 Aim and Objectives of the Study

Aim: The primary aim of this project is to design, implement, and evaluate a secure, user-friendly, and web-based system for reporting and managing social injustice cases at the University of Benin.

Objectives: To achieve this aim, the following specific objectives will be pursued:

To analyze the existing mechanisms and reporting culture for social injustice at the University of Benin and identify their specific limitations and challenges.

To design the system architecture for a secure, confidential, and anonymous web-based reporting platform, outlining its core modules, database structure, and user flows.

To develop a functional prototype of the system, using a Django back-end for robust case management and a React.js front-end for a responsive user interface.

To test the developed system's functionality, security, and usability in a controlled environment to ensure it meets the needs of both students (reporters) and administrators (case managers).

1.4 Research Questions

This study will be guided by the following research questions:

What are the primary challenges, inefficiencies, and cultural barriers of the current social injustice reporting system at the University of Benin, as identified through a survey of the student body?

What are the key functional and non-functional requirements for a web-based reporting system to be considered trustworthy, secure, and effective by students?

How can a system using Django and React.js be architected to ensure user anonymity, secure data handling, and efficient case management?

To what extent does the developed prototype improve the perceived confidentiality, efficiency, and user satisfaction of the reporting process compared to traditional methods?

1.6 Scope of the Study

This project encompasses the design, development, and local testing of a web-based prototype for reporting and managing social injustice incidents at the University of Benin (UNIBEN). The system’s key features include:

A secure user authentication module for administrators.

An anonymous reporting form for students or other reporters.

An authenticated User reporting form and user dashboard.

A back-end dashboard for administrators to manage and update case statuses.

A feature that allows reporters to track the status of their submitted cases anonymously.

In addition, the study includes a survey of students to assess the existing culture of reporting social injustice, including their awareness of reporting channels, willingness to report incidents, and perceived barriers to using current mechanisms. The survey results informed the system design, ensuring that the prototype aligns with student needs and expectations.

The study is limited to the development of a prototype and does not extend to native mobile applications for iOS or Android. Furthermore, the system will operate as a standalone application and will not be fully integrated with the University of Benin central student records database, ensuring data separation and enhanced confidentiality. The primary focus is on the system's technical functionality, usability, and alignment with student reporting practices, rather than its long-term sociological impact.

1.7 Limitations of the Study

Several limitations were encountered in the course of this study, which should be acknowledged:

Time Constraints: The project was conducted within the academic calendar, limiting the period available for extensive system development, multi-phase testing, and iterative refinements. Some advanced features and functionalities may be deferred to future work.

Sample Demographics: The survey assessing the culture of reporting social injustice was conducted primarily with a sample of University of Benin students. Staff and faculty perspectives were not included, which may limit the representativeness of the findings across the entire university community.

Access to Institutional Data: Access to official records and sensitive data on past social injustice reports was restricted due to confidentiality policies. As a result, the study relied on

student survey responses and prototype testing rather than a comprehensive comparison with historical data.

Deployment Environment: The system was developed and tested in a controlled, local environment. Its performance, security, and stability under high-volume or live deployment conditions were not fully assessed.

Infrastructure Limitations: The study acknowledges challenges such as intermittent internet connectivity and unstable power supply within UNIBEN. These factors may affect the performance and usability of the system in a live setting.

Financial and Technical Constraints: As a self-funded academic project, limited resources restricted the use of premium software, third-party security audit tools, and paid hosting for broader testing. Additionally, the researcher's individual coding and development expertise may have constrained the complexity and optimization of system features.

1.8 Organization of the Study

The project report is organized into five chapters, each addressing specific aspects of the study:

Chapter One: Introduces the study, presenting the background, statement of the problem, aim and objectives, research questions, significance, scope, and limitations.

Chapter Two: Reviews relevant literature, including theoretical frameworks on social injustice, existing reporting mechanisms in tertiary institutions, and digital systems for case management.

Chapter Three: Explains the research methodology, covering the system development life cycle (SDLC), system design models such as Use Case and Entity-Relationship Diagrams, and the technologies (Django and React.js) used for implementation.

Chapter Four: Details the system design and implementation, describing system architecture, functional modules, user interface, security features, and results from usability and functional testing.

Chapter Five: Presents the conclusion, summarizing key findings, evaluating how well the study achieved its objectives, and providing recommendations for system improvement, broader deployment, and future research.

1.9 Definition of Terms

To ensure clarity and understanding, the following key terms are defined as they are used in this study:

Social Injustice: Any act or practice within the university environment that results in unfair treatment, discrimination, harassment, bullying, or inequitable academic or administrative practices affecting students, faculty, or staff (Baba, 2024).

Web-Based System: A software application accessed through a web browser, hosted on a server, which allows users to perform tasks such as reporting, tracking, and managing cases digitally without installing local software (Edosomwan, 2024).

Anonymous Reporting: The process by which a user submits a complaint or report without revealing their identity, ensuring confidentiality and minimizing fear of retaliation.

Authenticated Users: Individuals who create accounts and log into the system to access additional features, such as submitting reports, tracking case status, and managing their profiles. Authentication ensures that user actions are secure and traceable while maintaining appropriate privacy controls.

Case Management: The systematic process of recording, tracking, monitoring, and resolving reported incidents of social injustice within an institution.

User Authentication: A security process that verifies the identity of a system user, ensuring that access to administrative or sensitive functions is restricted to authorized personnel.

Usability: The measure of how effectively and efficiently users can interact with the system, including ease of navigation, clarity of interface, and overall user satisfaction.

Prototype: A preliminary version of the web-based system developed to demonstrate functionality, gather user feedback, and guide further development.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter presents a critical review of existing theoretical and empirical literature on reporting mechanisms for social injustice within higher education institutions. It lays the academic groundwork for the development of a web-based system designed to enhance the way incidents of injustice are recorded, reported, and addressed. The review begins by discussing essential concepts such as social injustice, institutional grievance mechanisms, and reporting channels. It then explores theoretical models that explain reporting behaviour, evaluates system development methodologies, and examines technological solutions that facilitate secure and accessible reporting.

Contemporary research demonstrates that universities worldwide continue to struggle with enduring problems of harassment, discrimination and abuse of authority. Such misconduct is often unreported, largely because of fear of retaliation and scepticism about institutional response mechanisms. For example, a study of Nigerian higher education institutions found that 37 % of students reported experiencing sexual violence, yet many did not report formally due to lack of trust in the system (Ogunfowokan et al., 2024). At the same time, the growth of digital technologies especially secure web-based platforms presents a promising avenue to overcome these reporting barriers by enabling anonymity, increasing accessibility and improving case management (Baba, 2024; Ibitomisin & Oghenekaro, 2023).

The structure of this literature review is designed to create a comprehensive perspective on both the problem domain and potential solutions. It is organised as follows: first, conceptual and theoretical frameworks are clarified; next, empirical studies that document institutional responses to reporting challenges are examined; finally, the review highlights the research gap,

specifically the scarcity of cost-effective, anonymous, context-specific reporting systems for Nigerian universities.

2.2 Conceptual Framework: Social Injustice

In tertiary institutions, social injustice refers to systemic inequalities and the unfair treatment of individuals or groups based on identity, status, or power within academic settings. These injustices frequently manifest as discrimination, harassment, exclusion, or the denial of fundamental rights thereby undermining equity, access to opportunities, and the academic freedom necessary for learning and growth (Obi & Ziprebo, 2024; Shadows of Inequality, 2023).

Research in Nigeria highlights the magnitude of this phenomenon. A national survey of higher education students found that approximately 37% reported experiencing sexual violence, demonstrating a widespread pattern of harassment and abuse (Bello et al., 2024). Additionally, discrimination persists across gender and socio-economic lines, with 20–30% of female students reporting unwanted sexual advances or pressure for sexual favours from lecturers or peers (Shadows of Inequality, 2023).

These patterns indicate that social injustice in Nigerian universities is both interpersonal and institutional. Institutional injustices are reflected in biased access to resources, unfair evaluation or promotion practices, and the normalization of power asymmetries between staff and students or privileged and marginalized groups (Obalade & Mtembu, 2023; Olowu & Adebisi, 2022).

In summary, social injustice in the university context involves both the direct unfair treatment of individuals and systemic arrangements that entrench disadvantage and exclude meaningful participation. Conceptualizing injustice in this manner establishes a foundation for developing

reporting systems that address not only isolated incidents but also structural forms of unfairness.

2.2.1 Forms of Social Injustice in Nigerian Universities

Social injustice within Nigerian universities is tangible and manifests in multiple, overlapping forms:

Discrimination and Harassment: Individuals may be treated unfairly due to gender, ethnicity, religion, disability, or socio-economic status. Gender-based harassment remains particularly pervasive, with female students disproportionately affected. A national survey reported that nearly 28% of female undergraduates experienced sexual harassment from peers or lecturers, negatively affecting their academic engagement and mental well-being (Bello et al., 2024; Obande-Ogbuinya, 2022). Ethnic favoritism and religious discrimination have also been documented, creating hostile learning environments and reducing equitable access to opportunities (Obi & Ziprebo, 2024).

Exploitation and Academic Corruption: Abuse of authority occurs when lecturers or administrators leverage their positions for personal gain, often demanding sexual favours, bribes, or other benefits in exchange for grades or opportunities, a practice colloquially referred to as "sorting." Surveys indicate that approximately 15–20% of students report experiencing or witnessing such exploitation in Nigerian universities, highlighting a persistent culture of academic corruption that undermines trust and fairness (Okeke, 2021; Bello et al., 2024).

Exclusion and Marginalization: Certain groups, including students with disabilities, minorities, or those from low-income backgrounds, are often excluded from academic or social opportunities. This includes limited access to mentorship, research projects, and leadership roles. Research shows that students from marginalized groups are 25% less likely to participate

in university decision-making structures or receive academic support, reinforcing systemic inequities (Obalade & Mtembu, 2023).

Institutional and Systemic Injustice: These injustices are embedded in university policies, governance structures, and culture. Examples include biased grading systems, selective disciplinary action, and unequal distribution of resources favoring specific groups. Studies indicate that institutional bias significantly contributes to underreporting of grievances, as 62% of students who encountered harassment did not report incidents due to lack of trust in university processes (Olowu & Adebisi, 2022; Shadows of Inequality, 2023).

2.2.2 Classification of Injustice

The various forms of social injustice in Nigerian universities can be categorized according to their domain, illustrating the multi-layered nature of the problem:

Social Injustice: This encompasses gender-based violence, harassment, and unfair treatment that directly affect students' well-being and academic participation. National surveys report that about 28% of female students in Nigerian universities have experienced harassment or sexual misconduct, highlighting the urgency of addressing social injustice (Bello et al., 2024).

Economic Injustice: Economic disparities lead to unequal access to educational resources, scholarships, grants, and research opportunities. Students from lower socio-economic backgrounds often face barriers that reduce their competitiveness and limit academic advancement. Studies indicate that nearly 22% of students from low-income families feel disadvantaged in accessing academic opportunities compared to wealthier peers (Obi & Ziprebo, 2024).

Political Injustice: This relates to the exclusion of students or minority groups from meaningful participation in decision-making processes, including student unions, university committees, or policy consultations. Such exclusion undermines democratic representation and reinforces

power imbalances. Research shows that 35% of student organizations in Nigerian universities report underrepresentation of minority groups in leadership positions (Olatunji & Adebayo, 2022).

Cultural Injustice: Cultural marginalization occurs when institutions fail to recognize or respect the identities, languages, or traditions of minority groups. This form of injustice can affect students' sense of belonging and engagement. Approximately 18% of students from minority ethnic or cultural backgrounds report feeling excluded from academic and social programs due to institutional insensitivity (Obalade & Mtembu, 2023).

These classifications underscore that social injustice in Nigerian universities is not confined to interpersonal interactions but extends to systemic inequities affecting social, economic, political, and cultural domains. Any reporting system aimed at addressing these issues must therefore be capable of handling this complexity, ensuring that all forms of injustice are captured, reported, and addressed effectively.

2.3 Conceptual Framework: Reporting Systems

2.3.1 Defining Reporting Systems

A reporting system refers to a structured platform or method that enables individuals to formally document incidents, grievances, or misconduct for review and corrective action by an authoritative body (Adekunle & Yusuf, 2021). In the context of social injustice, these systems serve as crucial mechanisms for victims to seek redress and accountability while enabling institutions to respond effectively.

For a reporting system to be effective, it must be accessible, confidential, impartial, and transparent. Accessibility ensures that users can easily submit reports regardless of location or technical skill. Confidentiality protects the identity and safety of reporters, particularly in sensitive cases such as harassment or discrimination. Impartiality guarantees that all reports are

treated fairly, without bias, and transparency fosters trust by allowing users to track the progress of their complaints (Akinlade & Eze, 2022).

2.3.2 Typologies of Reporting Systems

Tertiary institutions have historically relied on a variety of reporting channels, which have evolved from traditional methods to digital platforms:

Manual or Face-to-Face Reporting:

This involves direct communication with designated authorities such as deans, student affairs officers, or the use of physical suggestion boxes. While familiar, these methods often lack confidentiality and discourage reporting of sensitive cases. For example, a 2022 study reported that over 60% of students avoided face-to-face reporting due to fear of retaliation or judgment (Adekunle & Yusuf, 2021).

Telephone Hotlines:

Hotlines provide a voice-based, confidential reporting option, often used in urgent scenarios. However, they require trained personnel and continuous monitoring, which can be resource-intensive. Additionally, these systems can struggle with tracking complaints and generating data for institutional decision-making (Akinlade & Eze, 2022).

Basic Digital Forms (Email/Web Forms):

These platforms allow users to submit reports remotely and attach supporting evidence. Despite this advantage, they frequently lack anonymity, secure storage, and structured case tracking, which can result in reports being overlooked or ignored. Studies in Nigerian universities indicate that nearly 40% of digital submissions via basic forms went unacknowledged, highlighting the need for more robust systems (Oladejo & Onwuzuruike, 2023).

Advanced Web-Based Platforms:

These platforms integrate secure submission portals, case management dashboards, automated routing, and status-tracking features. They are designed to support anonymous or confidential reporting while providing institutions with tools to manage complaints efficiently. Evidence from pilot studies in Nigerian universities demonstrates that such platforms improve response times by up to 50% and increase user trust and satisfaction (Ogunfowokan et al., 2023).

Mobile Applications:

Mobile apps offer the added benefits of real-time reporting, geotagging, and push notifications. However, their effectiveness can be limited by device compatibility, security vulnerabilities, and inconsistent internet access, particularly in rural areas or regions with limited connectivity (Ibitomisin & Oghenekaro, 2023).

2.4 Theoretical Frameworks

Several theories provide insights into why victims of social injustice often remain silent and what influences their reporting behavior. Understanding these frameworks is essential for analyzing reporting challenges in higher education institutions.

Procedural Justice Theory

Procedural Justice Theory emphasizes that individuals are more likely to engage with a system when its processes are perceived as fair, consistent, and transparent (Tyler, 2020). Research in African universities indicates that students often refrain from reporting harassment when they believe institutional procedures are biased or inconsistently applied. For instance, Maringe and Ojo (2021) reported that over 58% of Nigerian university students avoided submitting complaints due to mistrust in procedural fairness.

Organizational Justice Theory

Organizational Justice Theory extends the concept of procedural fairness to include distributive justice (fairness of outcomes) and interactional justice (respectful treatment during processes) (Colquitt et al., 2021). Evidence from Nigerian universities shows that victims often feel disrespected or dismissed during grievance processes, which discourages further reporting (Afolabi et al., 2021).

Critical Theory (Power Imbalance)

Critical Theory highlights how entrenched power dynamics such as lecturer-student or administrator-staff relationships can suppress reporting (Okeke, 2020). Fear of retaliation, academic penalties, or social exclusion deters students from reporting incidents. A recent survey indicated that about 62% of Nigerian students cited fear of reprisal as a major barrier to reporting harassment (WARDC, 2022).

Whistleblowing Theory

Whistleblowing Theory posits that individuals report wrongdoing when they:

feel morally obliged,

trust that the system will protect them from retaliation, and

believe reporting will lead to meaningful action (Near & Miceli, 2016).

In Nigerian universities, a study by WARDC (2022) found that 64% of students who experienced harassment did not report it, mainly because they doubted that their complaints would be addressed.

Social Ecological Theory

Social Ecological Theory explains that behaviour is influenced by multiple interconnected layers: individual, interpersonal, institutional, and cultural factors (Waller et al., 2020). In

Nigeria, cultural stigmas and societal norms significantly reduce the likelihood of reporting harassment (Eze et al., 2022).

2.5.1 Prevalence and Reporting Barriers in Nigeria

Evidence consistently shows that social injustice, particularly harassment, discrimination, and exploitation, is widespread in Nigerian universities, yet formal reporting remains critically low. Recent surveys indicate that over 60% of students who experienced harassment did not formally report incidents (Ogunfowokan, Adebowale, & Fasina, 2023). Mapayi, Fapohunda, and Ede (2023) similarly found that a majority of survivors refrained from reporting due to fear of negative consequences.

The primary barriers include:

Fear of Retaliation: Students often worry about academic victimization, delayed grading, or intimidation from staff and peers.

Mistrust of Institutional Processes: Many students believe that complaints will be ignored, mishandled, or result in unfair treatment.

Stigma – Victim-blaming attitudes and social judgment discourage students from speaking out.

Lack of Confidentiality: Concerns over identity exposure or leaks of personal information significantly reduce reporting (Mapayi et al., 2023; Ogunfowokan et al., 2023).

These findings highlight the urgent need for reporting systems that prioritize anonymity, confidentiality, and trust, providing students with secure channels to voice grievances without fear.

2.5.2 Technology Pilots and System Evaluations

Studies on digital reporting systems in Nigeria suggest that technology can help overcome reporting barriers. Ibitomisin and Oghenekaro (2023) evaluated a web-based complaint platform in a Nigerian university and found that it improved case logging speed, record-keeping accuracy, and user satisfaction compared to traditional, manual reporting methods. Users reported an increase in perceived safety and confidence, with nearly 70% indicating they were more likely to report incidents through the digital platform.

Despite these positive outcomes, research cautions that technology alone is insufficient. Institutional commitment, supportive policies, and staff training remain critical for ensuring that reports lead to meaningful action and accountability (Ibitomisin & Oghenekaro, 2023; Ogunfowokan et al., 2023).

2.6 Strategies for Enhancing Reporting Systems

Research on social injustice reporting emphasizes that technology alone cannot solve the problem of underreporting. Effective systems must combine security, awareness, support, and accountability to build trust and encourage usage. Key strategies identified in the literature include:

Anonymity and Confidentiality

Protecting the identity of reporters is fundamental to encouraging reporting, particularly in contexts where fear of retaliation is high. Evidence shows that when anonymity is guaranteed, reporting rates increase significantly. For example, studies in Nigerian universities indicate that over 65% of students are more likely to report harassment when confidentiality is assured (Cortina & Magley, 2021; Ogunfowokan et al., 2023). Secure online portals and anonymous hotlines are consistently cited as effective tools for mitigating fear and increasing participation.

Awareness and Education

Even the most secure system is ineffective if students are unaware of its existence. Awareness campaigns, orientation sessions, and regular communications ensure that students know how and where to report incidents. Bondestam and Lundqvist (2020) highlight that awareness interventions in universities increased reporting rates by approximately 40%, emphasizing the importance of visibility and education in fostering engagement with grievance mechanisms.

Independent Oversight and Transparency

Mistrust in institutional processes is a major barrier to reporting. Studies suggest that independent oversight bodies or clearly defined monitoring procedures improve user confidence. Transparent case-handling procedures and the ability to track the status of a report without revealing identities are shown to enhance trust, making users twice as likely to report incidents (Ibitomisin & Oghenekaro, 2023).

Psychosocial Support

Victims of harassment and discrimination require emotional and psychological support in addition to procedural remedies. Linking reporting systems to counseling services, peer-support programs, or mental health resources improves the overall effectiveness of the system and helps survivors cope with trauma (Cortina & Magley, 2021).

Accountability and Follow-Up

A reporting system must lead to tangible action; collecting cases without follow-up erodes credibility. Evidence from higher education settings shows that systems with visible accountability measures such as feedback loops and progress notifications significantly increase user trust and engagement (National Academies of Sciences, Engineering, and Medicine, 2022).

Collectively, these strategies highlight that successful reporting systems require a multi-faceted approach, combining secure technology with institutional commitment, awareness campaigns, and supportive resources to overcome barriers to reporting.

2.7 Synthesis and Identified Gaps

A review of the literature reveals important insights into social injustice reporting in Nigerian universities, including the University of Benin.

Persistent Problem: Social injustice including harassment, discrimination, exploitation, and systemic bias—is widespread in higher education institutions. Surveys indicate that between 60% and 70% of students in Nigerian universities, including the University of Benin, report experiencing some form of harassment or unfair treatment during their academic journey (Ogunfowokan et al., 2024; Mapayi et al., 2023). This shows the severity and pervasiveness of the problem.

Low Reporting Rates: Despite the high prevalence of social injustice, formal reporting remains extremely low. Many students avoid using institutional reporting channels due to fear of retaliation, stigma, mistrust of university authorities, and concerns about confidentiality (Mapayi et al., 2023; Ogunfowokan et al., 2024). Studies show that approximately 64% of students who experience harassment do not report incidents, highlighting significant barriers in existing reporting systems.

Potential of Technology: Digital solutions, particularly secure web-based platforms, have shown promise in improving reporting rates by offering anonymity, confidentiality, and efficient case management (Ibitomisin & Oghenekaro, 2023). Pilot studies conducted in Nigerian tertiary institutions demonstrate faster case logging, improved archival records, and higher user satisfaction compared to traditional face-to-face or manual reporting methods.

Need for Context-Specific Solutions: While global literature offers models for grievance reporting, few are tailored to the Nigerian context, particularly the University of Benin. Existing platforms often fail to account for local challenges such as hierarchical institutional structures, limited resources, and varying levels of internet access (Ogunfowokan et al., 2024; Ibitomisin & Oghenekaro, 2023).

Identified Gap: There is a clear need for research and development of a secure, anonymous, and affordable reporting system specifically designed for Nigerian universities. Such a system should address institutional culture, build student trust, and integrate mechanisms for transparency and accountability, ensuring that victims feel safe to report incidents.

Conclusion: Addressing these gaps requires the development of a reporting platform that is accessible, secure, and contextually relevant to the University of Benin. By combining anonymity, confidentiality, and ease of use, such a system can enhance reporting rates, promote institutional accountability, and ultimately support a safer and more equitable academic environment.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN METHODOLOGY

3.1 Introduction

System analysis is a careful and organized process that focuses on understanding how a system works, identifying its individual components, and examining how they interact to achieve specific goals. It provides the foundation for recognizing weaknesses, inefficiencies, and opportunities for improvement within a system.

This chapter therefore discusses the analysis and design process used in developing the proposed web-based social injustice reporting system. It explains the steps taken to gather user requirements, understand the existing challenges, and design a functional solution that supports anonymous reporting, efficient case management, and accessible communication between students and administrators.

3.2 Materials and Methods

This study adopted a structured quantitative research approach using a questionnaire administered to students of the University of Benin to assess their experiences with social injustice and the effectiveness of existing reporting mechanisms. The quantitative approach was chosen because it allows for the systematic collection and statistical analysis of measurable data, thereby ensuring objectivity and generalizability of findings (Creswell & Creswell, 2018). Using numerical data from surveys makes it easier to identify relationships and patterns across large groups of respondents, which is essential for evaluating social issues within a university context (Bryman, 2016).

Data Collection

Data were gathered through a structured online survey designed using Google Forms. The questionnaire consisted of both closed-ended and Likert-scale questions to capture participants' experiences of injustice, their awareness of reporting systems, and their trust in institutional responses. This method was selected because it facilitates wide participation, quick data aggregation, and efficient processing (Saunders et al., 2019).

Data Analysis Procedure

The collected data were exported to Google Sheets for organization and analysis. The analysis process involved several systematic steps:

Data Cleaning: Responses were screened for missing or inconsistent entries to ensure reliability. Duplicate or incomplete submissions were removed.

Data Coding: Categorical responses (e.g., gender, faculty, type of injustice experienced) were numerically coded to enable statistical evaluation.

Descriptive Analysis: Basic descriptive statistics such as frequency counts, percentages, and averages were generated to identify the distribution of responses and highlight dominant patterns.

Trend Identification: Responses were visually represented using bar charts and pie charts to identify major trends, such as common types of injustice reported and preferred reporting channels.

Interpretation: The analyzed data were interpreted in line with the research objectives to determine how students perceive social injustice reporting and the gaps within existing systems.

3.3 Sampling Method

A simple random sampling method was employed to select students from various academic disciplines, providing diverse perspectives on social injustice and reporting issues, and ensuring that the survey results accurately reflected the wider university student population.

3.4 Study Participants

The participants included 133 students from several discipline at the University of Benin

3.5 Study Instruments

The primary data collection instrument for this study was a structured questionnaire developed using Google Forms. It was designed to gather information from University of Benin students regarding their experiences, perceptions, and suggestions on social injustice within the campus. Prior to administration, the questionnaire was reviewed and validated by the project supervisor to ensure clarity, relevance, and content accuracy. This instrument provided critical insights into prevalent injustices, obstacles to reporting, and preferred design features for an effective and user-friendly digital reporting platform.

3.6 Questionnaire Design

The questionnaire used in this study was carefully designed to gather detailed information about students' experiences and perceptions of social injustice within the University of Benin. It consisted of both closed-ended and open-ended questions, allowing respondents to provide quantitative data as well as personal insights.

The questionnaire was divided into clear sections to ensure logical flow and ease of response. It began with general demographic questions and gradually progressed to more specific areas related to the study's objectives. The major sections included:

1. Demographics: Collected background information such as age range, gender, level of study, department, and place of residence (on-campus or off-campus) to analyze variations in experiences across different student groups.

2. Social Injustice Experience and Awareness: Explored the types, frequency, and responses to incidents such as discrimination, harassment, or intimidation.

3. Reporting Practices and Challenges: Examined students' willingness to report injustices and identified barriers such as fear of exposure, lack of trust, or inadequate support systems.

4. System Usability and Feature Evaluation: Collected feedback on preferred reporting methods, desired system features, anonymity, communication preferences, and confidentiality expectations.

5. Feedback and Suggestions: Provided space for additional comments and recommendations.

3.7 Data Distribution and Collection

Data collection for this study was conducted through online means to enhance accessibility and ensure broad participation. The survey instrument, developed using Google Forms, was distributed to University of Benin students via widely used social media platforms, including WhatsApp, Facebook, and Telegram.

Prior to participation, respondents received an informed consent statement outlining the study's purpose, voluntary nature, and measures to protect confidentiality and anonymity in accordance with standard ethical research practices. The use of digital platforms facilitated efficient, cost-effective data collection from a diverse student population.

3.8 Data Analysis

This section presents the analysis of survey data obtained from 133 students of the University of Benin. The objective was to examine students' experiences with social injustice and

reporting practices on campus, as well as their expectations for an effective web-based reporting system. The analysis provides critical insights into students' attitudes, behaviors, and perceptions regarding unfair treatment, harassment, discrimination, and existing reporting mechanisms within the university context. These findings are essential for informing the design and functionality of the proposed system, ensuring it effectively addresses students' real challenges. The results are presented using tables, charts, and graphs to facilitate clear interpretation and support evidence-based system development.

3.8.1 Demographic Data

Table 3.1: Age of Participants

Age	Number of Entries	Frequency (%)
Under 18	20	15
18-24	79	59.4
25-30	33	24.8
Above 30	1	0.8

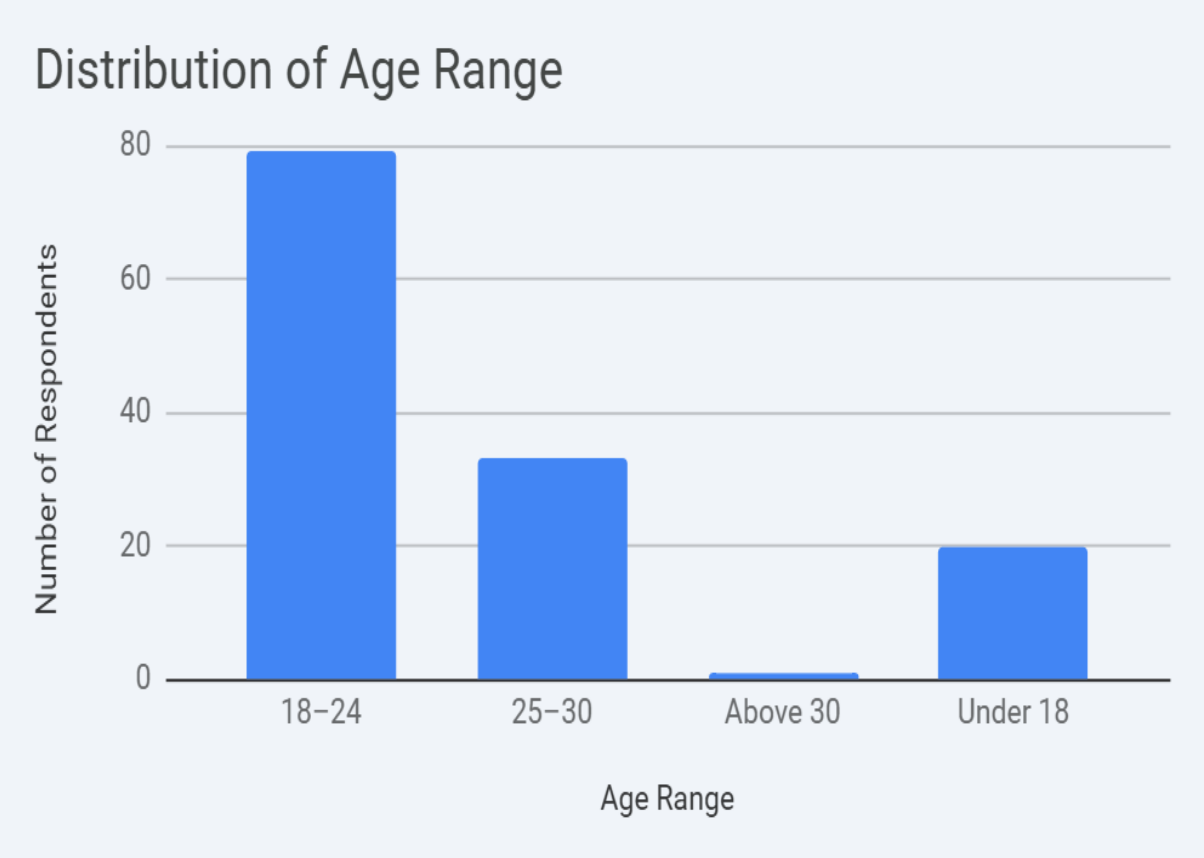


Figure 3.1: Chart showing The Age of Participants

From Table 3.1, most respondents (59.4%) fall within the 18–24 age range, 24.8% of participant fall between the age range of 25-30, 15% of participant were under 18, while less than 1% (0.8%) were Above 30

Table 3.2: Gender of Participants

Gender	Number of entries	Frequency (%)
male	53	39.8
Female	74	55.6
Prefer not to say	6	4.5

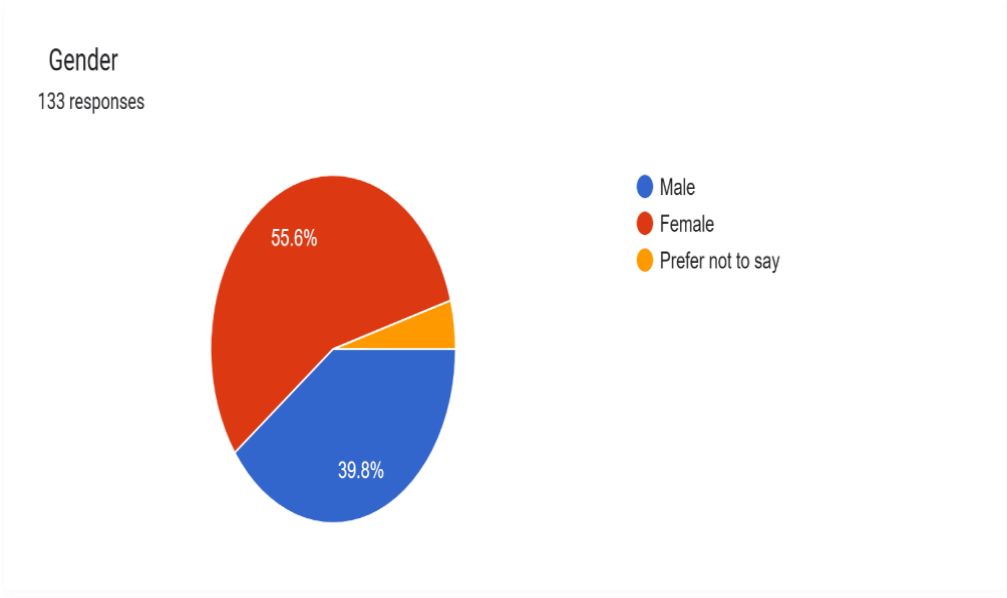


Figure 3.2: Chart Showing Gender Distribution

From figure 3.2, 55.6% Of the participants were female while 39.8% were male. 4.5% preferred not to identify their gender

Table 3.3: Level of Study

Level of study	Number of entries	Frequency (%)
100	15	11.3
200	18	13.5
300	19	14.3
400	45	33.8
500	12	9
600	24	18

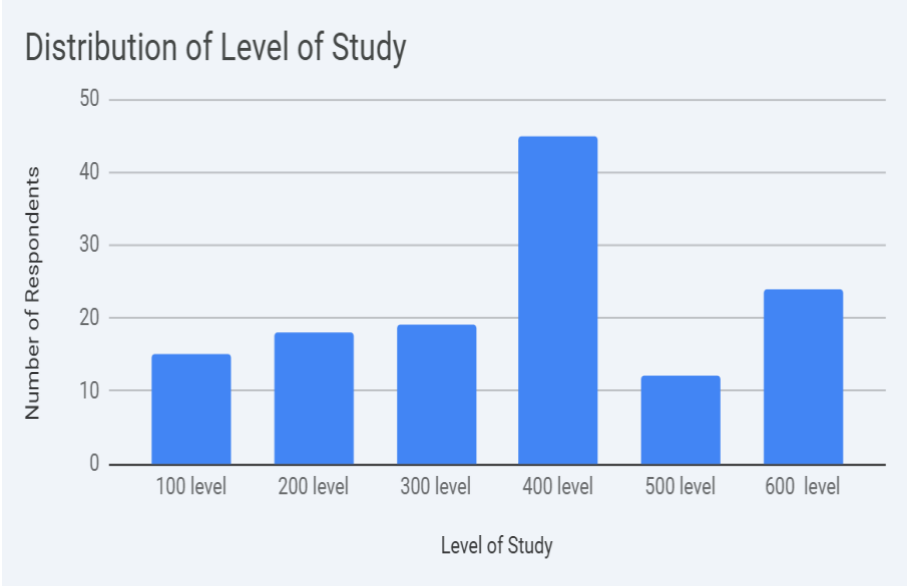


Figure 3.3: Chart Distribution of Level of Study

Table 3.3: Level of Study shows the distribution of respondents according to their academic level. The data indicates that the majority of participants (33.8%) were in 400 level, followed by 600 level students (18%). Students in 300 and 200 levels made up 14.3% and 13.5% respectively, while 100 level students accounted for 11.3%. The least represented group were 500 level students (9%).

Table 3.4: Residence

Residence	Number of Entry	Frequency (%)
On campus	96	72.2
Off campus	37	27.8

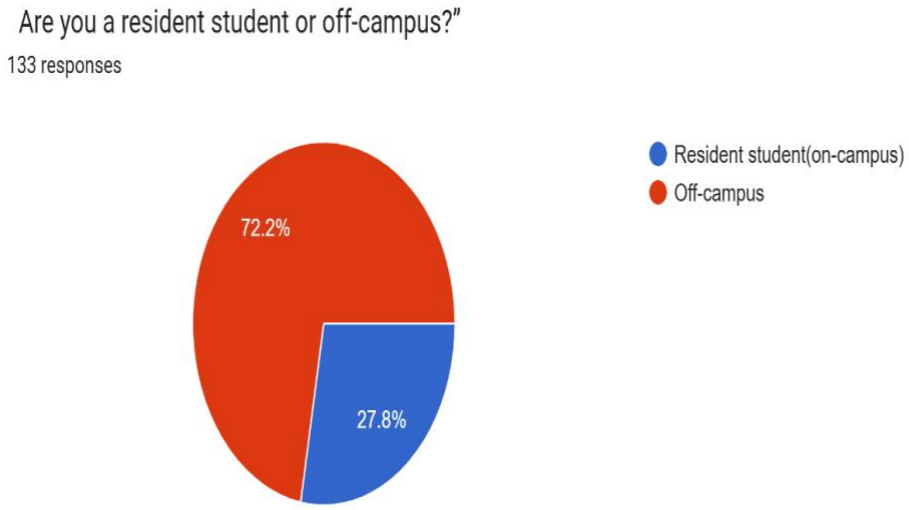


Figure 3.4: Chart showing residence distribution

3.8.2 **Awareness and Experience of social injustice**

Table 3.5: Number of those who have experienced social injustice

Experience/ witnessed	Number of Entries	Frequency%)
Yes, I have experienced it personally	76	57.1
Yes, I have witnessed it happen to others	28	21.1
NO	29	21.8

Over half of the respondents (57.1%) personally experienced social injustice, while 21.1% witnessed it happen to others. Only 21.8% reported no such experience. This shows that social injustice is a prevalent issue among students.

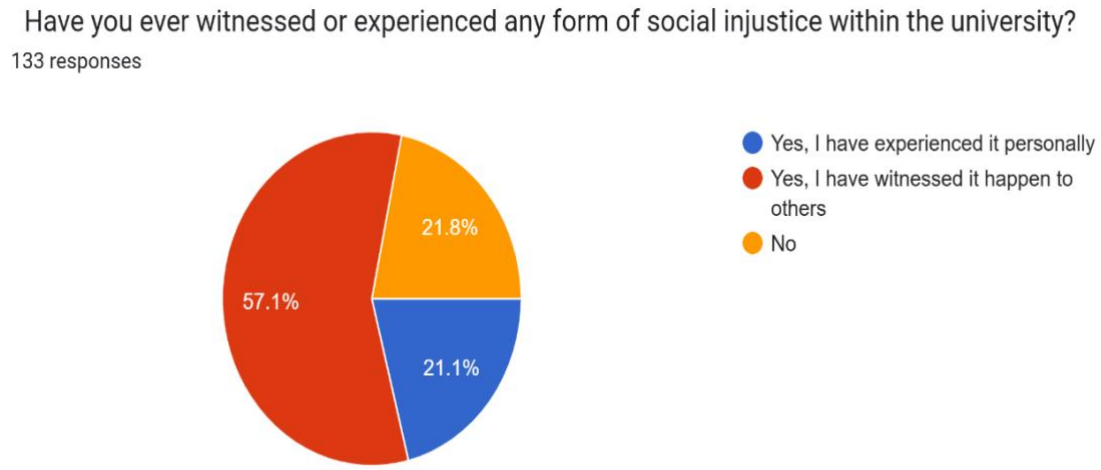


Figure 3.5: Chart showing Distribution of social injustice experience

Table 3.6: Types of injustice experienced

Types	Number of Entries	Frequency (%)
Sexual harassment	23	21.1
Gender discrimination	24	24
Academic favouritism or unfair grading	55	50.5
Bullying or intimidation	50	45.9
Racial or ethnic discrimination	8	7.3
Religious discrimination	9	8.3
Financial exploitation or bribery	44	40.4
others	5	4.6

From Table 3.5, The most common forms of injustice reported were academic favouritism or unfair grading (50.5%) and bullying or intimidation (45.9%). Other notable issues included financial exploitation or bribery (40.4%) and gender discrimination (24%), while racial and

religious discrimination were less frequent. This indicates that academic and interpersonal injustices are the most pressing concerns among students.

Figure 3.6: Chart showing Common types of injustice experienced

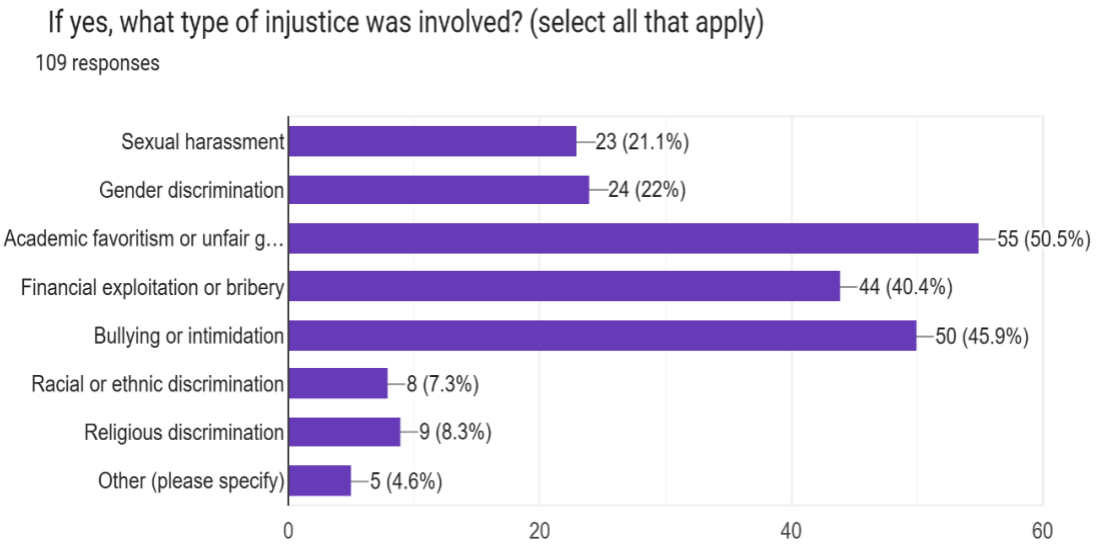


Table 3.7: Frequency of incidents

Frequency of occurance	Frequency(%)
Very often	30.8
Sometimes	42.1
Rarely	9.8
Never	8.3

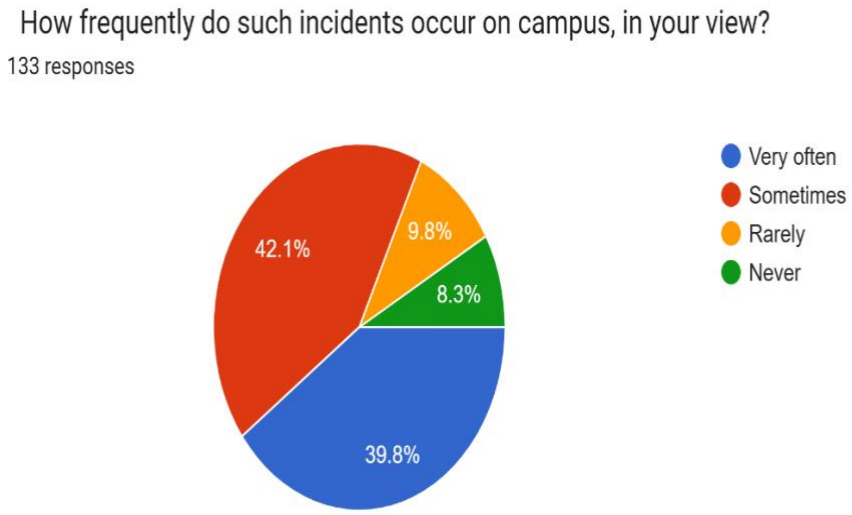


Figure 3.7: Chart S Frequency of incident.

The findings show that injustice incidents occur sometimes (42.1%) and very often (30.8%), indicating they are relatively common. Only a small portion of respondents reported experiencing them rarely (9.8%) or never (8.3%), highlighting a persistent pattern of recurring injustices within the institution.

Table 3.8: Tables showing how students rate the university's current effort at handling issues of social injustice

Effort rating on a scale of 1-5	Number of entries	Frequency (%)
1 (low)	34	25.6
2	44	33.1
3	32	24.1
4	15	11.3
5(high)	8	6

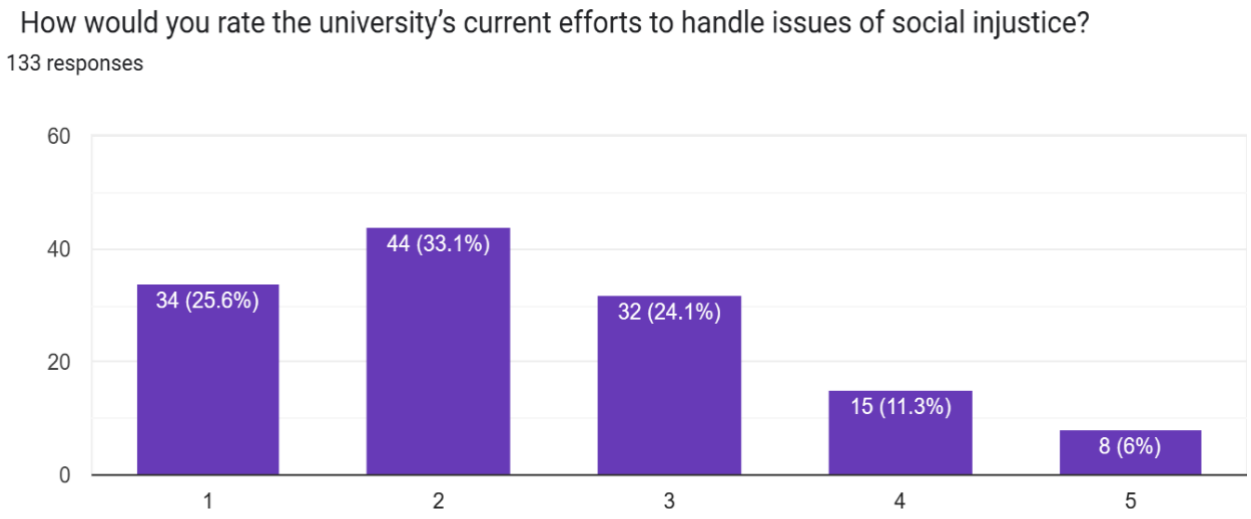


Figure 3.8: Chart showing how students rate the current effort at handling issues of social injustice

Findings from Table 3.7 indicate that the majority of students (58.7%) rated the university's effort at handling social injustice as poor (ratings 1–2). Only 17.3% rated the institution's effort positively (ratings 4–5), while 24.1% gave a neutral score of 3. This suggests that students generally perceive the university's response to social injustice as inadequate, with many believing that recurring injustices are not being effectively addressed.

3.8.1 Reporting Practices and Challenges

Table 3.9: Table showing the Frequency of reporting

Reported an injustice	Number of entries	Frequency (%)
No	86	82.3
yes	18	17.3

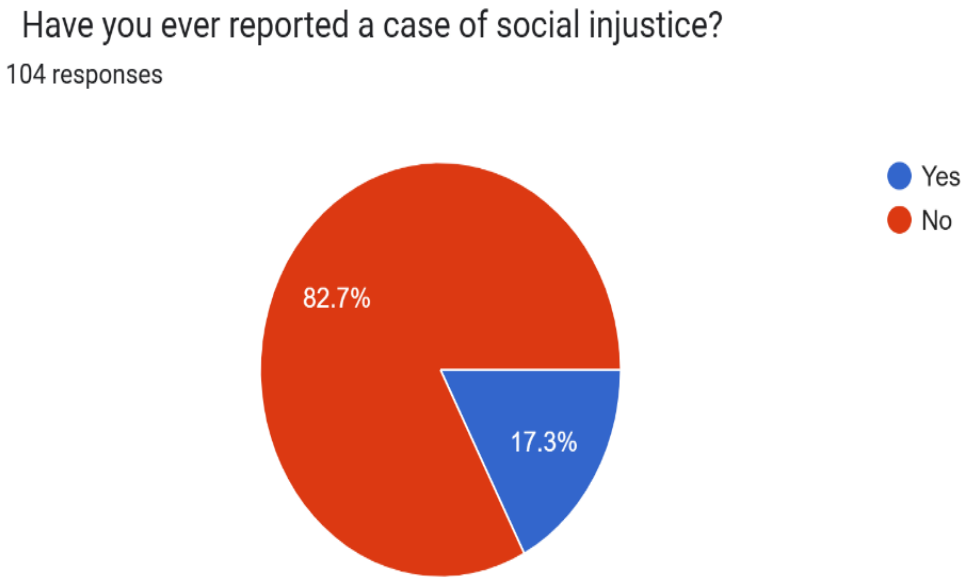


Figure 3.9: Chart showing frequency of reporting injustice

The results show that 82.3% of respondents did not report incidents of social injustice, while only 17.3% did.

Table 3.10: Table showing reasons for not reporting

Reason	Number of entries	Frequency (%)
Fear of retaliation or punishment	43	48.9
Lack of trust in the reporting system	50	56.8
No idea where or how to report	30	34.1
Concern about confidentiality	28	31.8
Belief that nothing will be done	43	48.9
Others	1	1.1

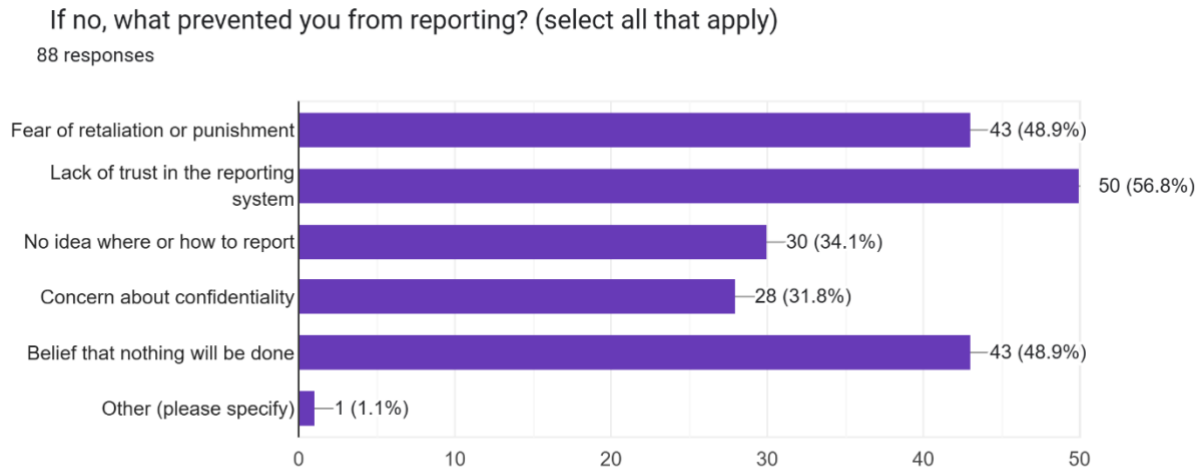


Figure 3.10: Chart showing reasons not reporting.

The findings from figure 3.8 reveal that the main reasons for low reporting rates include a lack of trust in the reporting system (56.8%), fear of retaliation or punishment (48.9%), and the belief that no action will be taken (48.9%). Additionally, 34.1% of respondents had no idea where or how to report, and 31.8% were concerned about confidentiality, highlighting the need for a more transparent and supportive reporting mechanism.

Table 3.11: Table showing the knowledge of an existing reporting channel

Knowledge of existing reporting system	Number of entries	Frequency (%)
yes	24	23.1
no	66	63.5
maybe	14	13.5

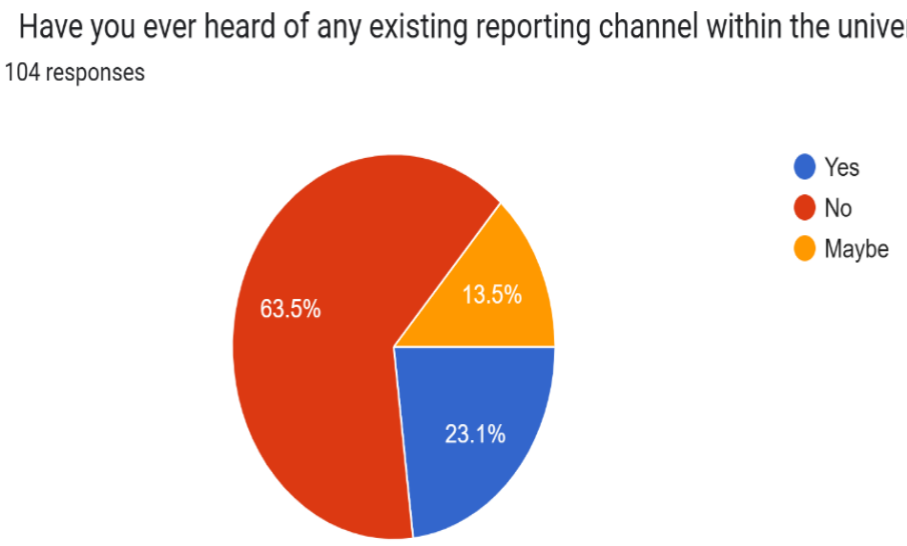


Figure 3.11: chart showing of existing reporting channel

The results from figure 3.9 indicate that a majority of respondents (63.5%) have no knowledge of any existing reporting system, while 23.1% acknowledged awareness, and 13.5% were uncertain. This highlights a significant information gap and lack of visibility regarding available reporting mechanisms within the institution.

Table 3.12: Table showing willingness to use an anonymous digital reporting platform

On a scale of 1-5	Number of entries	Frequencies (%)
1 (low intrest)	5	4.8
2	1	1
3	19	18.3
4	30	28.8
5 (high intrest)	49	47.1

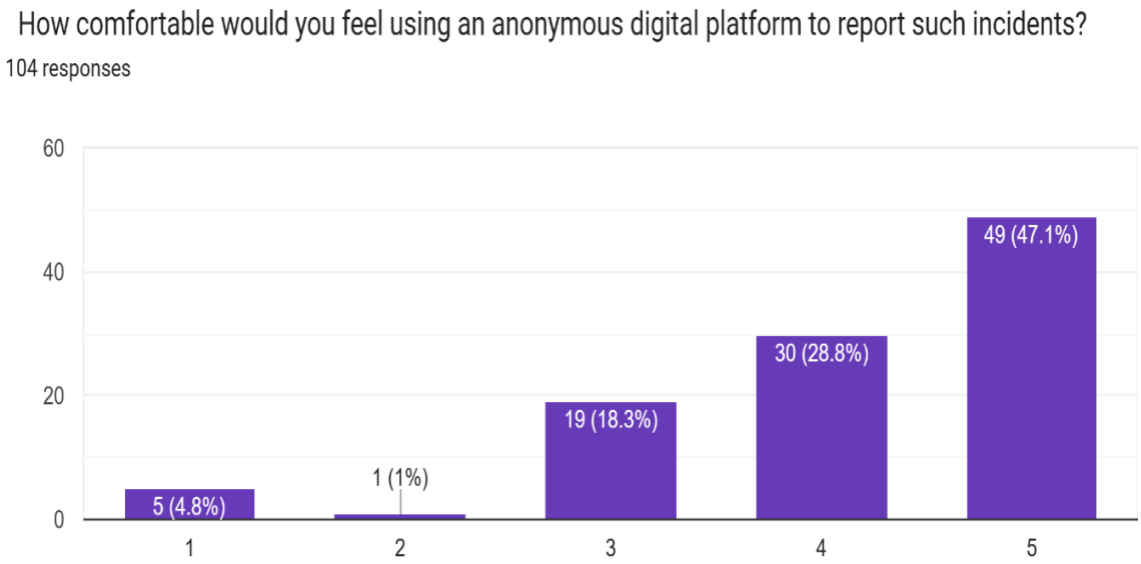


Figure 3.12: chart showing willingness to use an anonymous digital reporting platform

The results indicate a high willingness among students to use an anonymous reporting system, with 47.1% showing strong interest (rating 5) and 28.8% moderately interested (rating 4). Only a small fraction (5.8%) expressed low interest, suggesting that most students are open to engaging with a confidential, user-friendly reporting platform for social injustice cases.

Table 3.13: Table showing key features most important of a reporting system

Feature	Number of entries	Frequency(%)
Option for anonymous reporting	67	64.4
Progress tracking of reported cases	52	50
Email or SMS notifications for update	26	25
Public display of resolved cases(anonymized)	43	41
Easy-to-use interface	40	38
Data confidentiality and privacy	54	51

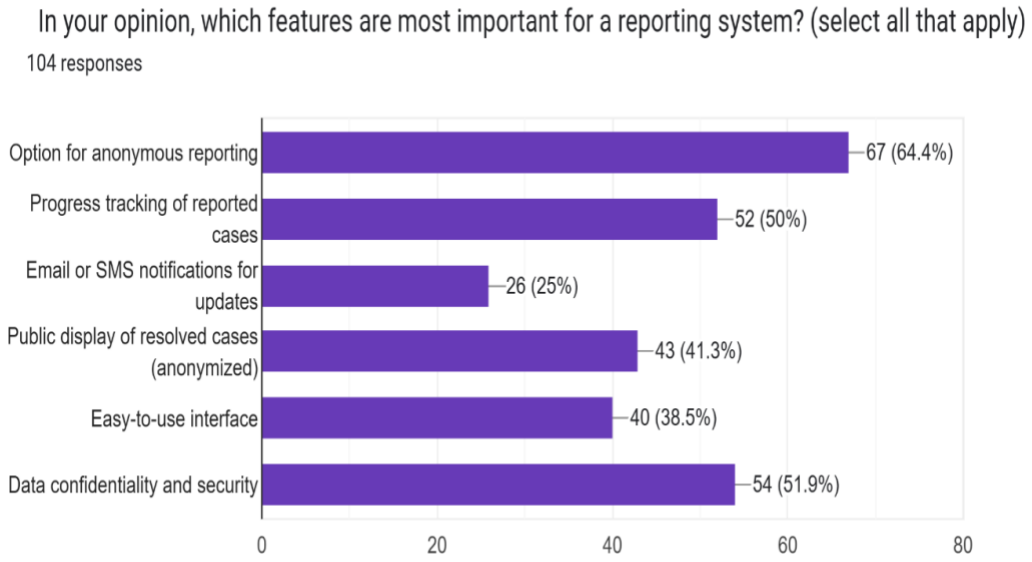


Figure 3.13: chart showing features most important for a reporting system

Findings from Table 3.10 reveal that the most valued feature in a reporting system is the option for anonymous reporting (64.4%), followed by data confidentiality and privacy (51%) and progress tracking of reported cases (50%). Other notable preferences include public display of resolved cases (41%) and an easy-to-use interface (38%), showing that students prioritize trust, transparency, and usability in a reporting platform.

Table 3.14: Table showing change in trust level if the university decided to use a transparent online reporting platform

Increased trust level	Number of entries	Frequency(%)
Yes, definitely	63	60.6
maybe	35	33.7
no	6	5.8

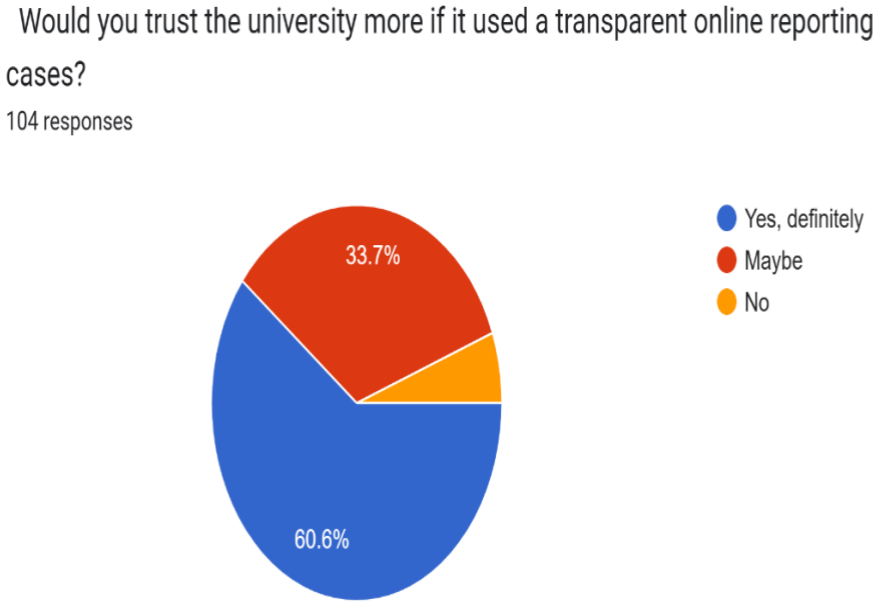


Figure 3.14: chart showing the change in trust if the university uses an online reporting system

Table 3.12 shows that a majority of students (60.6%) believe their trust in the university would increase if a transparent online reporting platform were implemented, while 33.7% remain uncertain and only 5.8% express doubt. This indicates that introducing a transparent digital system could significantly enhance students' confidence in the institution's handling of social injustice cases

3.8.3 System Usability and Feature Evaluation

Table 3.15: Table showing Features participant considered most important for the web-based system

Feature	Number of entries	Frequency (%)
Anonymous reporting	87	64.5
Case tracking/ Status update	56	42.1
Evidence uploading (pictures, videos, document)	60	45.1
Email notification	23	17.3
Public display of resolved cases	47	35.3
Monthly statistics/report	36	27.1
Live chat	34	25.6

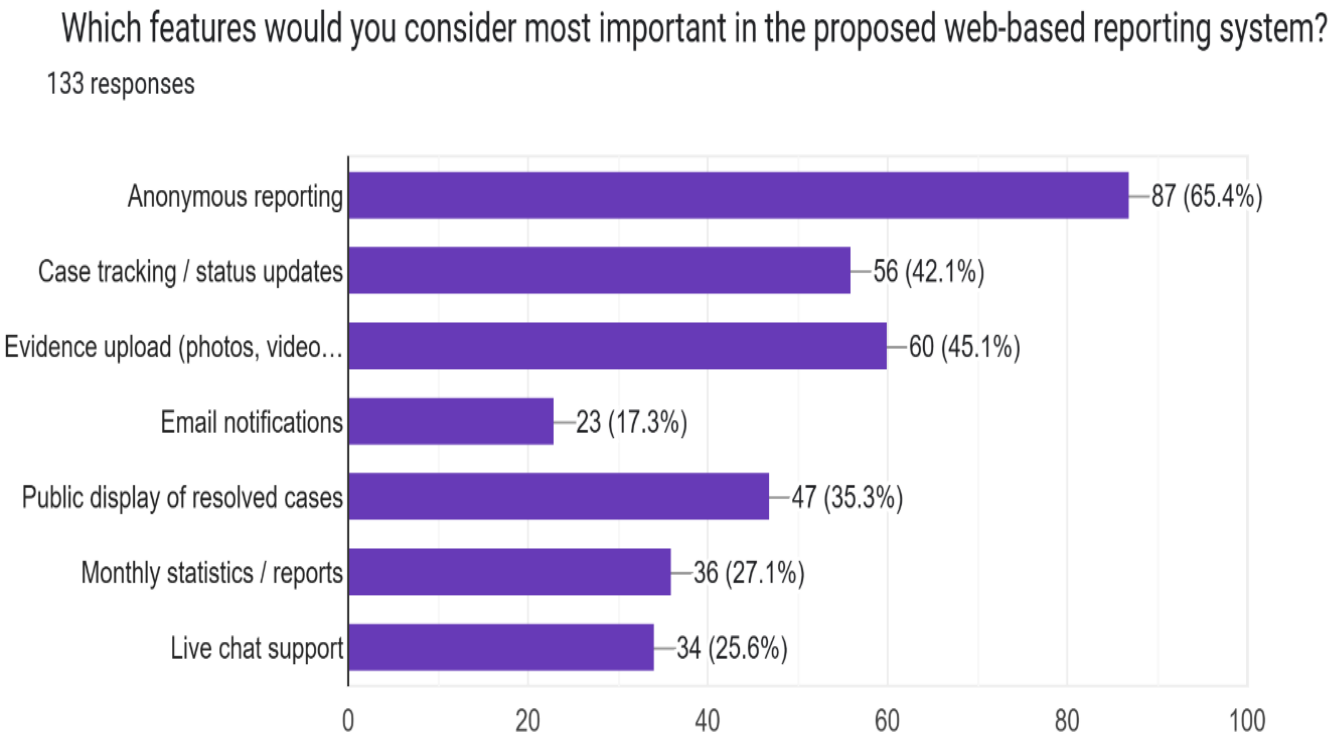


Figure 3.15 Chart showing features desired in the proposed web-based reporting system

Table 3.14 reveals that anonymous reporting (64.5%) is considered the most important feature by participants, followed by evidence uploading (45.1%) and case tracking or status updates (42.1%). Other features such as public display of resolved cases (35.3%), monthly reports (27.1%), and live chat (25.6%) also received notable support. This suggests that students prioritize privacy, transparency, and accountability in the design of the proposed web-based reporting system.

Table 3.16 Table showing User interface design preference

User interface design preference	Number of entries	Frequency (%)
Easy navigation	85	63.9
Clean and minimalist design	44	33.1
Bright and colorful theme	14	10.5
Professional / formal layou	42	31.6
Mobile-responsive design	51	38.3

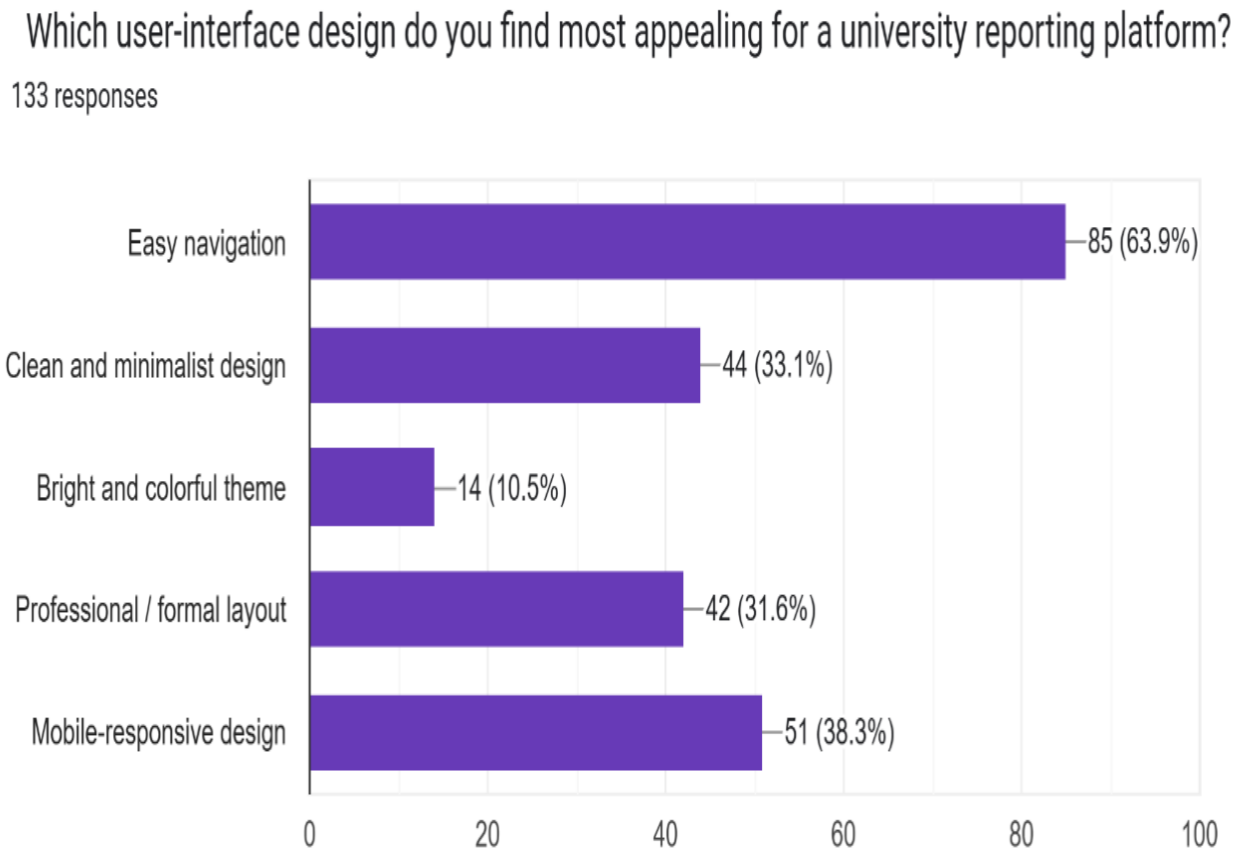


Figure 3.16 Chart showing user interface design preference

Table 3.15 indicates that easy navigation (63.9%) is the most preferred user interface feature among respondents, emphasizing the need for a platform that is simple and intuitive. This is followed by preferences for mobile-responsive design (38.3%), clean and minimalist layout (33.1%), and professional or formal appearance (31.6%). Meanwhile, only 10.5% of respondents favored a bright and colorful theme, suggesting that students prefer a user-friendly, professional, and functional interface for the reporting platform.

3.9 Overview of the Proposed System

The proposed system, SpeakUp, is a web-based social injustice reporting platform developed to address incidents of harassment, discrimination, and related forms of social misconduct within tertiary institutions, particularly the University of Benin. The platform is designed to promote a fair, inclusive, and accountable academic environment by offering students a secure, accessible, and transparent channel for reporting social injustice cases.

SpeakUp is developed using Django for the backend and React.js for the frontend, a combination that ensures a scalable, reliable, and user-friendly system. This architecture allows the platform to efficiently manage sensitive data while maintaining high standards of performance, integrity, and usability.

The system incorporates the following core functionalities:

1. Anonymous Reporting: Allows students to submit reports confidentially without disclosing their identity, encouraging openness and protecting their privacy.

2. Authenticated Reporting: Provides an option for users who prefer to identify themselves by logging in with their official university email for verified reporting and easier case follow-up.

3. Evidence Submission: Supports uploading of relevant materials such as photos, videos, or documents that can serve as evidence for reported incidents.

4. Case Management: Equips administrators with an interactive dashboard for viewing, assigning, and tracking cases throughout the investigation and resolution process.

5. Public Display of Resolved Cases: Publishes anonymized case summaries and statistics to promote institutional transparency and accountability.

7. Monthly Analysis and Reporting: Generates and exports analytical reports summarizing monthly trends and case distributions to support data-driven policy and decision-making.

8. User Dashboard: Provides registered users with a personalized interface for viewing their submitted reports, monitoring progress, and receiving timely updates.

Through these integrated features, SpeakUp aims to strengthen students' trust in the university's response mechanisms, enhance administrative efficiency, and foster a safer and more equitable academic community.

3.10 Proposed System Architecture

The architecture of the proposed SpeakUp platform, is designed to provide a secure, efficient, and user-centered environment for both reporters and administrators. It adopts a client–server model, which facilitates clear separation between the presentation layer (frontend) and the data management layer (backend), thereby improving scalability, maintainability, and overall system reliability.

The frontend of the system is developed using React.js, a JavaScript library known for building dynamic, responsive, and modular user interfaces. Through this interface, users can submit reports of social injustice, attach multimedia evidence (such as images, videos, and documents), and monitor the progress of their cases. The component-based architecture of React enhances code reusability and maintainability, while its responsive design ensures

compatibility across various devices. React also manages the system's application state using React Hooks and the Context API, providing a seamless and interactive user experience.

The backend is implemented using the Django REST Framework (DRF), which serves as the core of the application's logic and data processing. Django handles key backend operations including authentication, authorization, database management, validation of user inputs, and communication with the frontend through secure RESTful APIs. The framework's built-in security features, such as protection against cross-site scripting (XSS), SQL injection, and cross-site request forgery (CSRF), help safeguard sensitive information. Additionally, Django manages the processing of uploaded evidence files, automated email notifications, and generation of analytical reports to support administrative decision-making.

The database layer is powered by SQLite, a lightweight and serverless relational database engine integrated with Django. It stores essential data such as user profiles, reported incidents, uploaded files, case updates, and administrative logs. SQLite is particularly suitable for development and small-to-medium-scale deployment due to its simplicity, reliability, and ease of configuration. Moreover, it provides an easy migration path to more robust database systems such as PostgreSQL or MySQL when future scalability is required.

The authentication structure of the Speak Up platform supports two modes of operation: anonymous and authenticated reporting. The anonymous mode allows users to submit reports without providing personal information, thereby encouraging individuals to come forward without fear of identification or retaliation. The authenticated mode enables users to log in with their verified university email addresses, allowing for report tracking, follow-up communication, and accountability during investigations.

To streamline administrative operations, the system includes an administrator dashboard, implemented using Django Admin integrated with React components. This dashboard provides a comprehensive interface for viewing, assigning, and tracking reported cases. Administrators can update case statuses, record internal comments, and generate monthly statistical reports that highlight reporting patterns and trends. A super administrator oversees all operations, assigning cases to subordinate administrators to ensure accountability and efficient workload distribution.

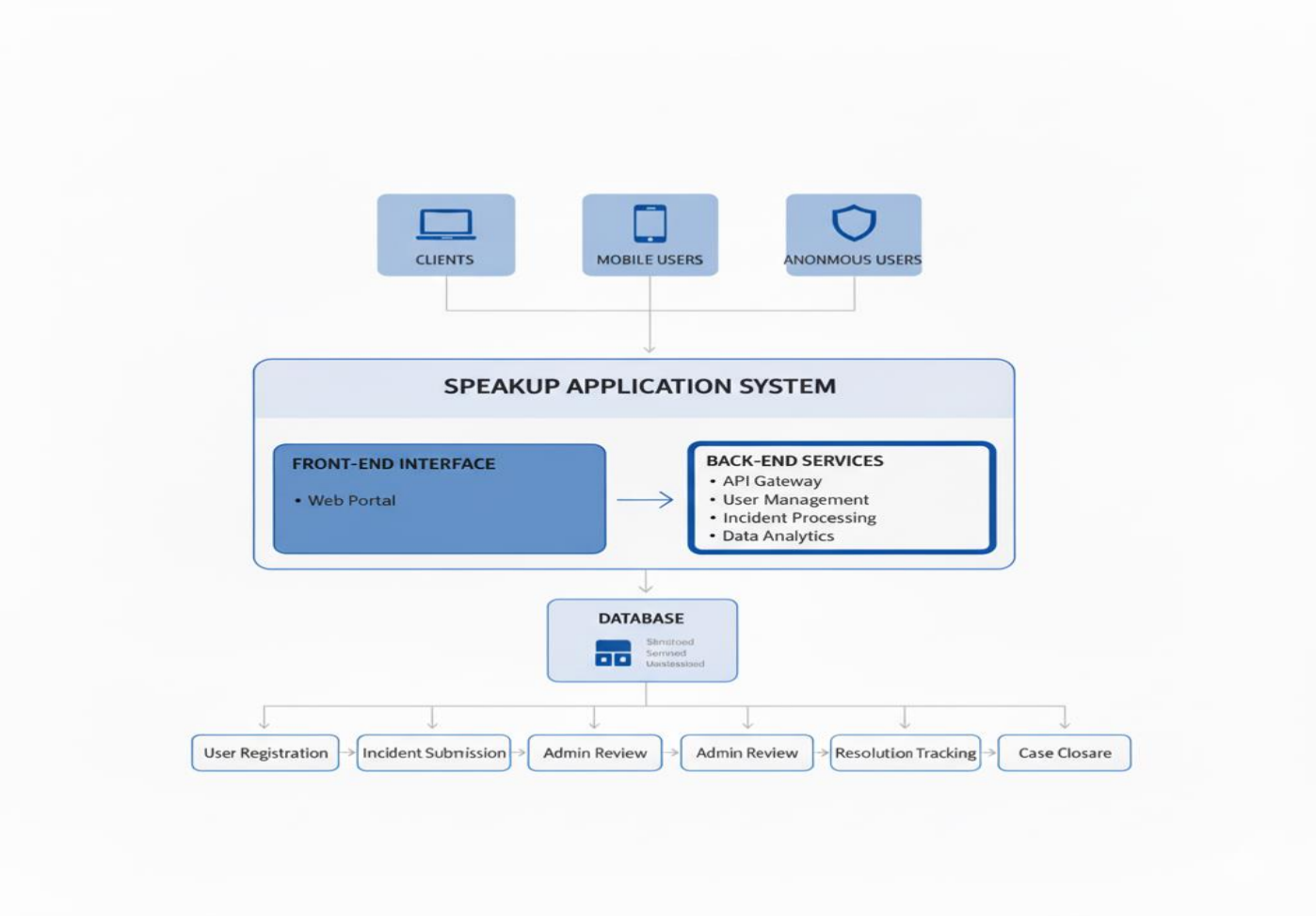


Figure 3.17 Proposed System Architecture Diagram

3.11 System Design

System design refers to the process of developing the architectural framework that defines how different components, modules, and interfaces of a system interact and function together to

achieve defined objectives. It outlines the system's overall structure, organization, and data flow, serving as a blueprint that guides implementation and ensures that all parts work cohesively to meet user and functional requirements.

3.12 System Design Tools

System design tools are software platforms that assist designers, engineers, and system architects in creating, analyzing, and documenting system structures. These tools help visualize system architecture, model component interactions, trace requirements, and promote collaboration among development teams. Common examples include:

Unified Modeling Language (UML): UML is a standardized visual modeling language used to describe and design the structure and behavior of systems. It provides graphical notations and conventions for representing system components, including use case, class, and state machine diagrams,12each serving a unique role in illustrating different aspects of the system.

Data Flow Diagram (DFD): A DFD visually depicts how data moves within a system, showing the relationships between processes, data stores, and external entities. It highlights how data is created, processed, stored, and transmitted, helping designers understand system functionality and data dependencies.

Entity-Relationship Diagram (ERD): An ERD is a graphical model that illustrates entities within a system and the relationships between them. It represents how data elements interact through various relationship types, such as one-to-one, one-to-many, and many-to-many connections, providing a clear view of data organization and linkage.

3.12.1 System Design Tool: UML

Unified Modeling Language (UML) is a standardized modeling language that uses a collection of diagrams to help system and software developers specify, visualize, construct, and document

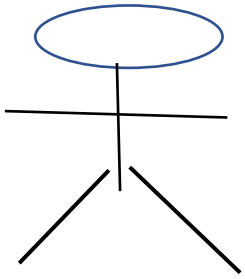
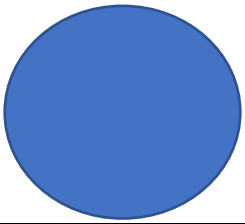
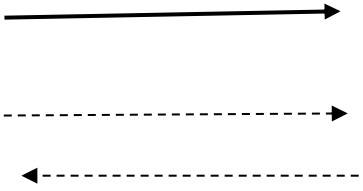
software systems. It provides a visual framework that enhances the understanding, design, and implementation of complex systems, improving clarity and maintainability. UML uses graphical notations to describe system components, their structure, and interactions. These notations are organized into different types of diagrams, each serving a specific purpose in representing the system. In this study, UML is used to model the SpeakUp Platform, helping to illustrate how different users and system component interact:

3.12.2 UML Use Case Diagram

A Use Case Diagram in UML visually represents how users interact with a system by showing the main actions or tasks (use cases) and the entities (actors) that perform them. It provides an overview of the system's functionality from the user's perspective, highlighting who performs specific actions and how they relate to each other.

In the Speak Up Platform, the use case diagram demonstrates the interactions between three main actors: Students, Admins, and the Super Admin. Students can create and submit reports on social injustices, upload supporting evidence (text, images, or audio), track report status, and provide feedback after resolution. Admins are responsible for reviewing and managing reports, updating case statuses, and providing resolution feedback. The Super Admin oversees the platform's overall operation, assigns cases to admins, monitors system performance, manages user accounts, and reviews analytics on reported issues.

Table 3.17 Use Case Notations and Descriptions

Object	Symbols	Description
Actors		External entities such as students, admins, or the Super Admin that interact with the system.
Use case		Actions or tasks that each actor performs in the system, such as submitting, managing, or reviewing reports.
System		Represents the SpeakUp Platform boundary, containing all use cases and actor interactions.
Relationship		Define connections between actors and use cases. Association: Direct interaction between actors and use cases. Include: Indicates that one use case includes another's functionality. Exclude: Indicates a use case omits another's functionality under certain conditions.

The use case diagram illustrates the different ways key actors (Anonymous Users, Registered Users, and Administrators) interact with the SpeakUp platform. Anonymous Users can submit incident reports and upload evidence without needing to create an account or log in, allowing them to report injustices confidentially. Registered Users, on the other hand, are required to

register and log in to access extended functionalities such as submitting verified reports, tracking report progress, viewing past submissions, and managing their profiles. Administrators are responsible for overall system supervision, which includes viewing and updating all reports, assigning cases to appropriate personnel, adding case notes, managing user accounts, and utilizing analytical dashboards. Their role ensures that the entire reporting and resolution process operates efficiently and transparently.

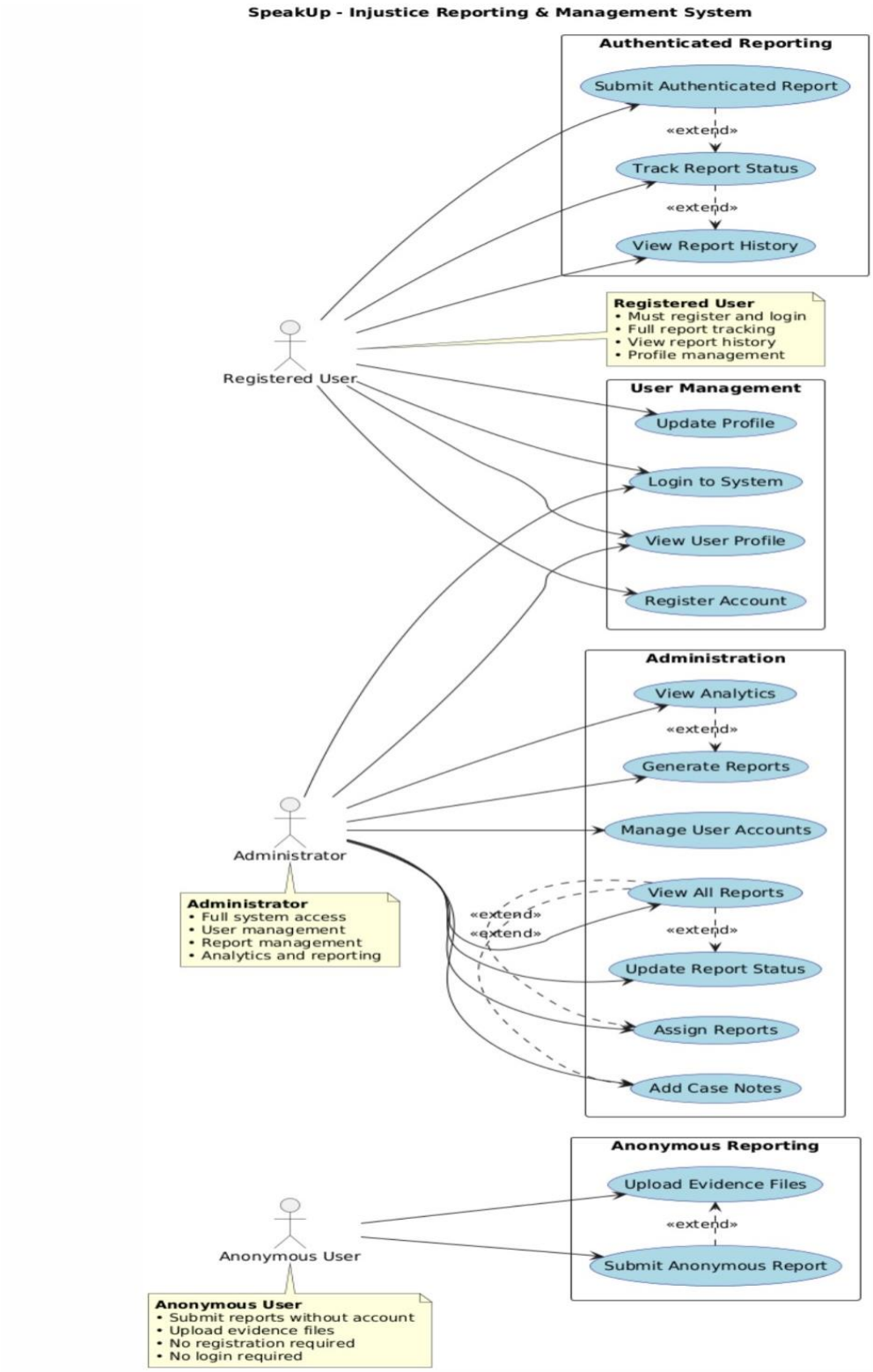


Figure 3.18 Case diagram

3.12.3 Entity Relationship Diagram

The Speak Up Platform employs an Entity Relationship Diagram (ERD) to illustrate the logical structure and relationships between key entities involved in the injustice reporting and users, administrators, and reports within the system. management process. This diagram provides a clear visualization of how data flows between users, administrators, and reports within the system.

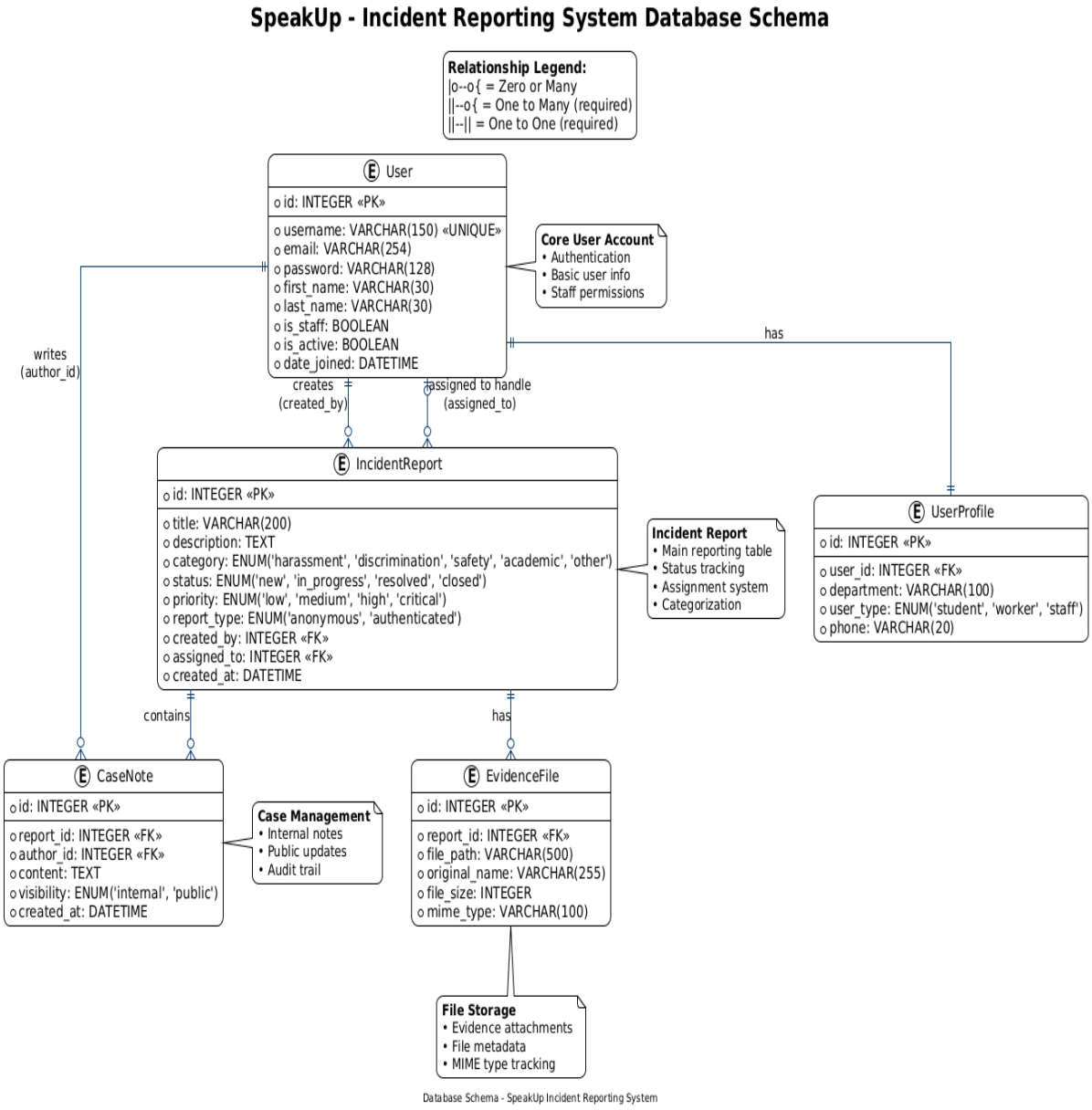


Figure 3.19: Entity Relationship Diagram of the Speak Up Platform

the User entity represents individuals who interact with the Speak Up system, including students, staff, and community members. Each user is uniquely identified by a User ID and contains essential authentication information such as an email address and password. This core entity is extended by a User Profile entity, linked in a one-to-one relationship, which stores additional demographic and organizational details like department affiliations and user type classifications (e.g., student, staff). Users can submit incident reports, track their report status, manage personal information, and receive feedback or updates on administrative actions.

Administrators are a specific type of user with higher privileges, responsible for overseeing and managing the entire reporting process. Each administrator has an Admin ID and defined permissions that grant them access to a comprehensive dashboard. Their responsibilities include verifying, categorizing, and assigning reports to appropriate units, monitoring case progress, ensuring resolution, and recording internal case notes.

The Incident Report entity serves as the core component of the system, representing submitted cases of injustice or misconduct. Each report is uniquely identified by a Report ID and contains comprehensive details, including a description, category, location, submission date/time, priority level, and current status. The system supports both anonymous and authenticated reporting modes.

To support each case, the Evidence File entity manages all supporting documentation and multimedia. It is linked to a specific Incident Report through a one-to-many relationship, ensuring a proper evidence chain-of-custody while handling various file types and metadata. Concurrently, the Case Note entity facilitates communication and documentation throughout the resolution lifecycle. Also linked to an Incident Report in a one-to-many relationship, this entity captures all status updates, internal administrative comments, and user-visible feedback, building a detailed case history.

These entities are interconnected through clear relationships. A one-to-one relationship links each User to their User Profile. A one-to-many relationship exists between a User and the Incident Reports they submit. The administrative workflow is supported by a flexible relationship that allows Administrators to be assigned to multiple reports, ensuring efficient workload distribution.

The ERD design enforces strict data integrity and security through role-based access control. Regular users can only access and modify their own reports and profiles, while administrators have broader system visibility. Through this comprehensive relationship structure, the system maintains complete audit trails, as all evidence and case notes are permanently linked to a report, ensuring accountability. Referential integrity is enforced via foreign key constraints, which prevents orphaned records (like evidence without a report) and maintains data consistency.

This ERD serves as a foundational blueprint for the database design, illustrating how entities such as users, administrators, and reports interact. It ensures data consistency, supports efficient case tracking, and facilitates the secure management of sensitive information across all stages of reporting and resolution.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

This chapter presents the implementation phase of the Speak Up platform, describing how the system was transformed from a design prototype into a fully functional application. It highlights the tools, programming languages, frameworks, and platforms used during development, along with the steps taken to ensure smooth deployment and user experience. The implementation process was guided by the need to create a secure, efficient, and user-friendly web-based system for reporting and managing social injustice cases within tertiary institutions.

4.1 Software Implementation Tools

The successful development of the SpeakUp web-based application required a combination of modern programming tools and technologies that ensured efficiency, scalability, and ease of maintenance. The selected tools and platforms provided an integrated environment for front-end and back-end development, database management, and deployment.

1. Implementation Language

TypeScript:

TypeScript was adopted as the primary programming language for the front-end development of Speak Up. It is a statically typed superset of JavaScript that enhances code readability, reduces runtime errors, and supports large-scale application development. The inclusion of static typing and interfaces helped maintain a structured and modular codebase, which simplified debugging and improved overall maintainability.

2. Implementation Frameworks

Frameworks provide pre-defined structures and reusable components that accelerate development and promote consistency. The following frameworks were utilized in building the Speak Up system:

a. Tailwind CSS:

Tailwind CSS, a utility-first styling framework, was used for designing the application's user interface. Its pre-defined utility classes allowed for rapid styling and easy customization without the need for extensive custom CSS. This ensured that the platform maintained a professional, responsive, and accessible design suitable for institutional use.

b. Django Framework:

For the back-end, Django was selected due to its robustness, built-in security features, and ability to handle complex database operations efficiently. Written in Python, Django follows the Model-View-Template (MVT) architectural pattern, which made it suitable for managing the logical flow of data between users, administrators, and the database. Django's authentication system and built-in admin interface were particularly useful for managing user roles and case reports securely.

c. ReactJS:

ReactJS was employed for the development of the front-end components of the web platform. Its component-based architecture allowed the creation of reusable and interactive user interfaces. By handling the view layer of the application efficiently, React improved performance and responsiveness during user interactions.

3. Development Environment

Visual Studio Code (VS Code):

Development was carried out using Visual Studio Code, a versatile source-code editor developed by Microsoft. It provided an integrated workspace with features such as syntax highlighting, debugging, intelligent code suggestions, and Git version control. VS Code also supported project scalability through extensions that facilitated easier integration with frameworks like Django and React.

4. Backend and Database Infrastructure

SQLite Database:

During development, the system utilized SQLite as the primary database due to its simplicity and ease of setup in local environments. It provided sufficient functionality for storing user profiles, incident reports, evidence files, and administrative feedback without requiring complex configuration.

Django ORM (Object Relational Mapper):

Django's ORM was employed to handle database operations. It allowed developers to interact with the database using Python objects instead of raw SQL queries, ensuring data consistency and security across all transactions.

5. Deployment Platform

For the current phase, the Speak Up system is deployed and running locally for testing and evaluation purposes. This allows developers to identify and resolve bugs before public deployment. However, future deployment plans include hosting the system on platforms such as Vercel or a university-managed server, ensuring wider accessibility, improved data security, and institutional ownership.

4.2 System Testing

System testing is a crucial phase in the implementation of the Speak Up platform. It involves a complete evaluation of all integrated modules to ensure that they function cohesively and meet both functional and performance requirements. This stage validates the system's reliability, usability, and efficiency in handling real-world operations related to reporting and managing social injustice cases within the university community.

4.2.1 Purpose of System Testing

The main objective of system testing was to verify that every component of the Speak Up platform performs as intended and aligns with the system's design and user specifications. This process ensures that features such as report submission, user authentication, administrative control, and feedback handling operate smoothly, without errors or inconsistencies.

4.2.2 Testing Approach

Manual testing techniques were employed to simulate real-world user interactions across various modules of the application. Test cases were developed for both students (reporters) and administrators to ensure that every process, from reporting an incident to managing and resolving cases, was accurately executed. Each test scenario focused on validating data flow, interface responsiveness, and overall system stability under typical usage conditions.

4.2.3 Modules Tested

User Authentication: The login and logout processes were tested to ensure that both users and administrators could securely access the platform. Password encryption, authentication validation, and session management were verified to maintain account security and prevent unauthorized access.

Incident Reporting: Testing confirmed that users could accurately report incidents by filling in required details such as category, description, and location. The evidence upload function was

validated to ensure users could attach supporting files such as images or videos to strengthen their reports.

Administrative Dashboard: The admin dashboard was tested to confirm its ability to provide an overview of all submitted cases, including their categories, status, and assigned handlers. The filtering and sorting features were evaluated to ensure administrators could easily navigate and manage large volumes of reports.

Incident Management: This aspect was tested to confirm that administrators could efficiently update case statuses (e.g., pending, under review, resolved) and track progress over time. The notification system was also reviewed to ensure that users received timely updates on their submitted cases.

User Dashboard: This aspect was tested to confirm that the user could track the progress of submitted case.

Case Management (Public and Private Comments): The case management module was tested to verify that both administrators and users could add comments to specific reports. The system distinguishes between public comments, visible to both users and administrators, and private comments, reserved for internal administrative discussions. This ensures effective communication during investigation and resolution while maintaining confidentiality where necessary.

Media Upload: Testing was conducted to ensure that uploaded media such as photos, videos, and documents were stored securely and properly linked to their corresponding reports. File validation checks for size and format were also confirmed to prevent system errors or malicious uploads.

Feedback Management: The feedback mechanism was tested to verify that users could submit reviews or suggestions about the handling of their reports. This promotes accountability, encourages transparency, and provides data for improving administrative response processes.

Monthly Analytical Reports: This feature was tested to ensure that the platform could automatically generate monthly summaries of all submitted incidents. These summaries included statistical breakdowns by category, resolution rate, and response time, providing administrators with valuable insights for performance evaluation and institutional decision-making.

4.3 Screenshots of Running Systen

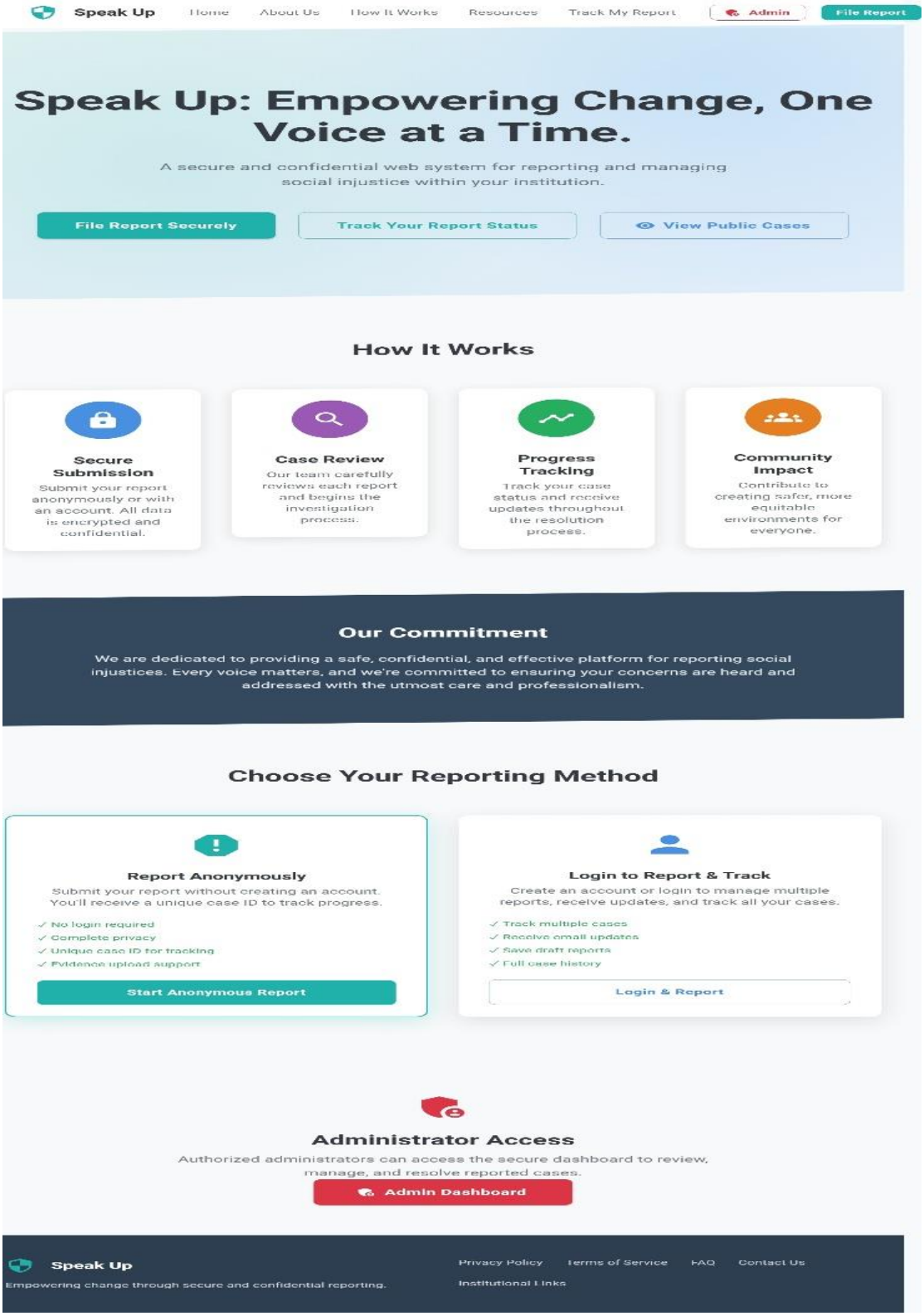


Figure 4.1 Homepage

The homepage, as illustrated in Figure 4.1, serves as the initial interface users encounter upon accessing the SpeakUp web application. It features a user-friendly and intuitive layout designed to facilitate seamless navigation. The homepage provides distinct options for reporting incidents either anonymously or as an authenticated user, ensuring accessibility and inclusivity. Additionally, users can easily access publicly available resolved cases through a dedicated navigation button. Administrative users are granted access to the system's management interface via the "Admin Login" option located at the top of the page or through the "Admin Login" button positioned at the bottom section. Furthermore, the "Track Your Case" feature enables authenticated users to conveniently monitor the status and progress of their submitted reports directly from their personalized dashboard.

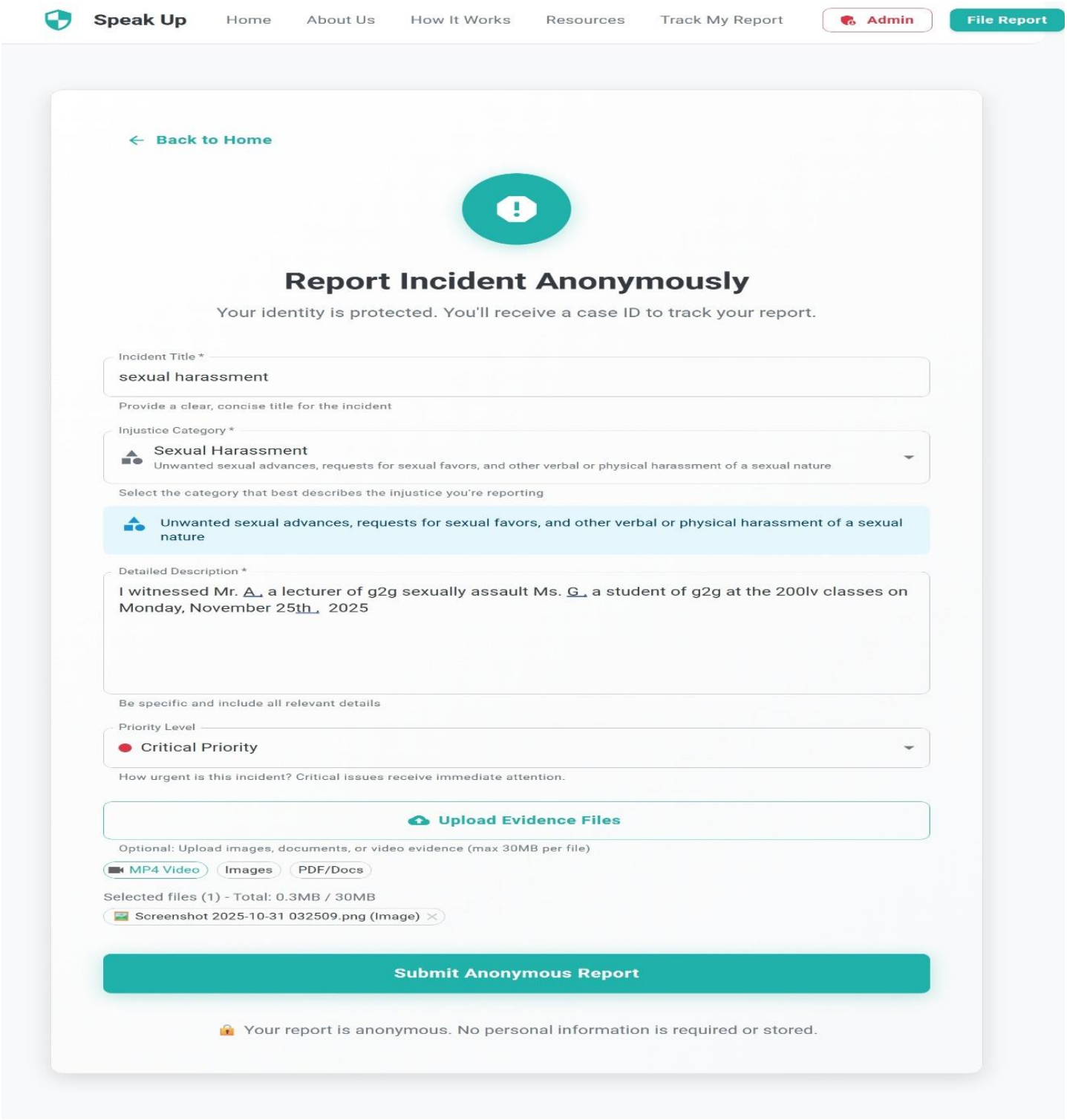


Figure 4.2 Anonymous Reporting page.

Figure 4.2 shows the reporting form, which prompts users to enter details including the Incident Title, Injustice Category, Detailed Description, and Priority Level. A dropdown menu allows users to select the type of injustice, accompanied by a brief definition to aid

understanding. The Upload Evidence field enables users to attach relevant files, such as images, videos, or documents, with clear file-type and size restrictions displayed.

At the bottom of the form, a prominent “Submit Anonymous Report” button allows users to finalize their submission. A concluding privacy message reinforces user anonymity, assuring that no personal data is required or stored. The overall design ensures accessibility, clarity, and trustworthiness, which are critical for encouraging honest reporting.

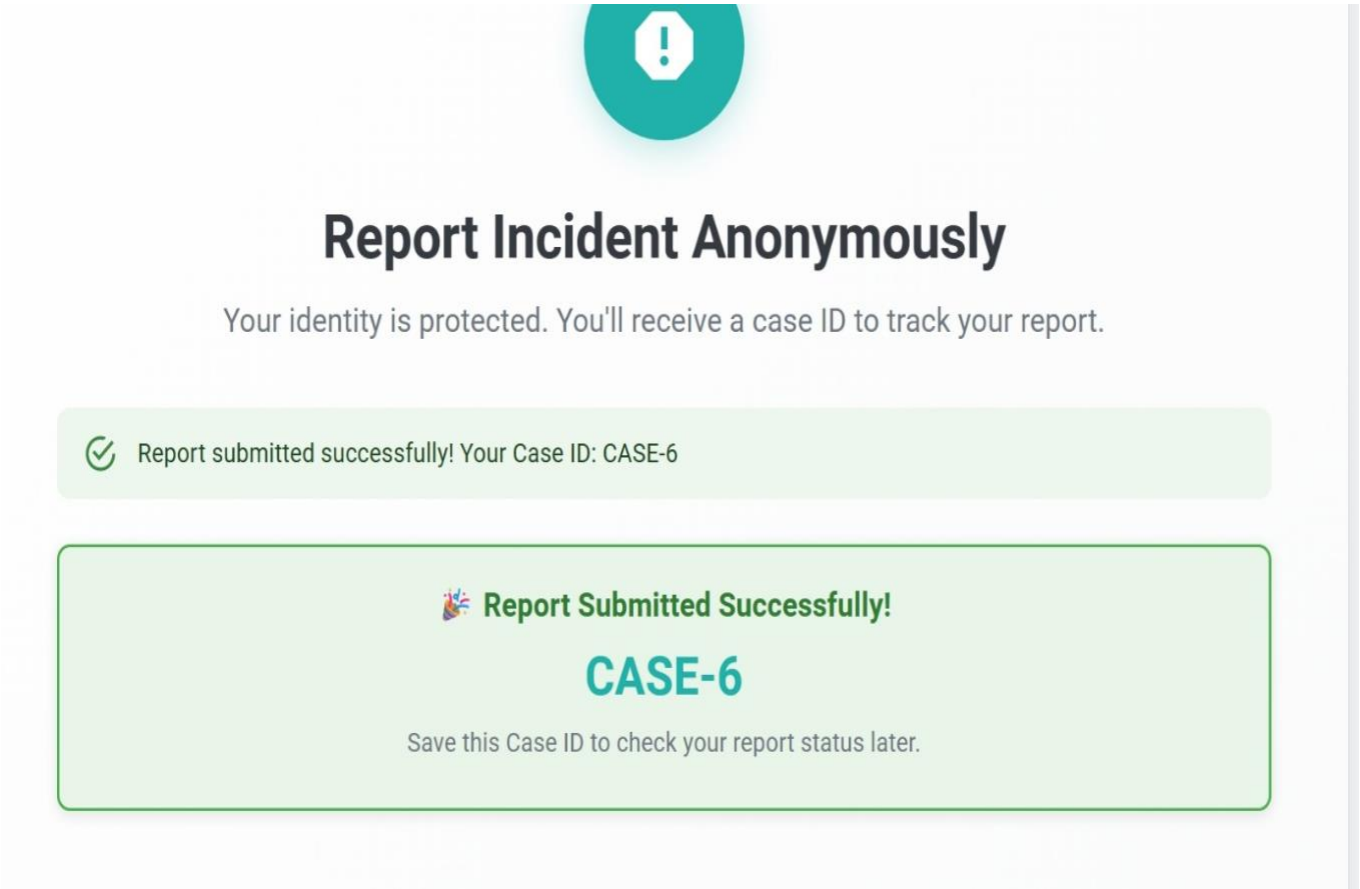


Figure 4.3 Successful Anonymous report

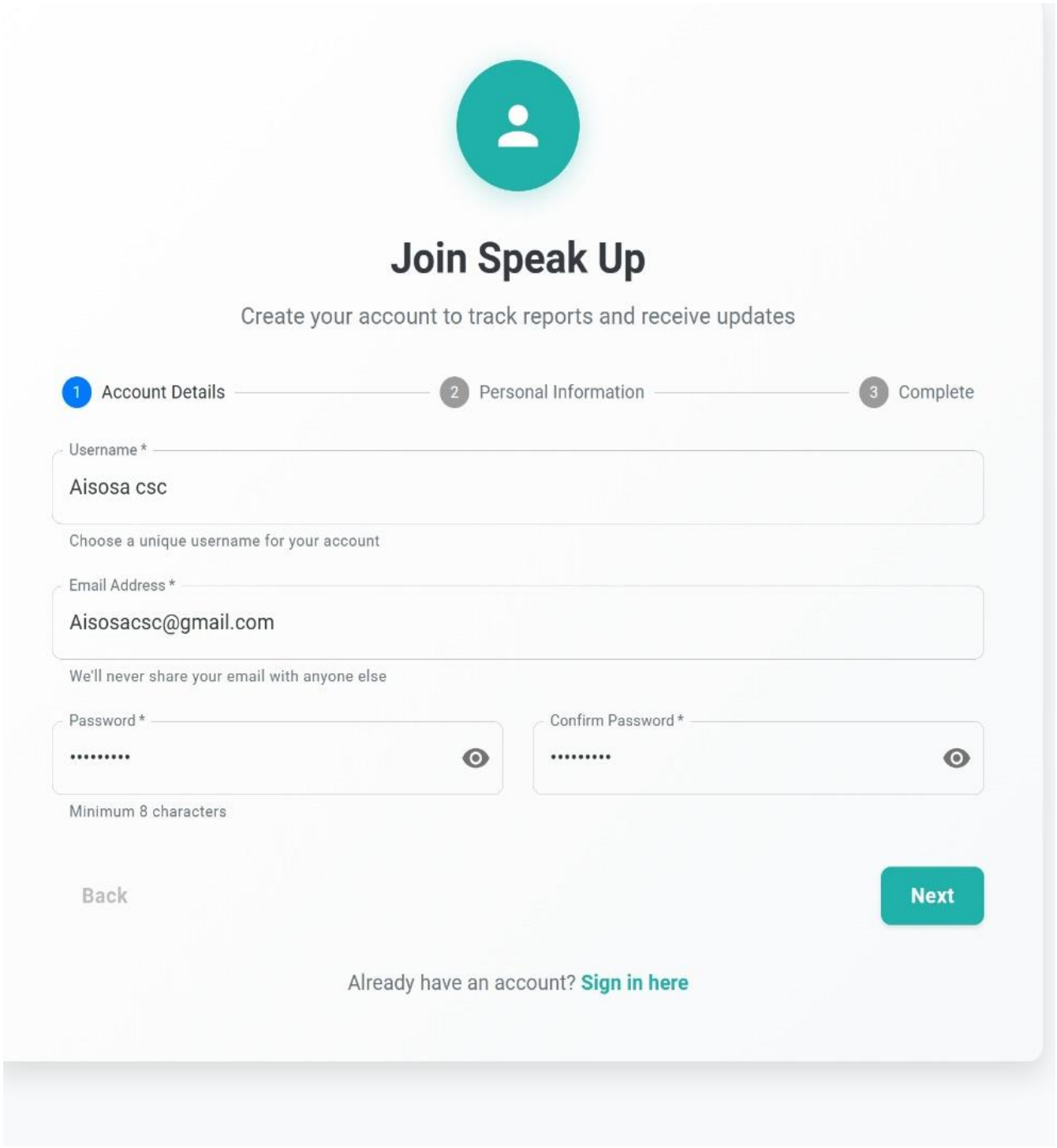


Figure 4.4 Authenticated User Registration page.1

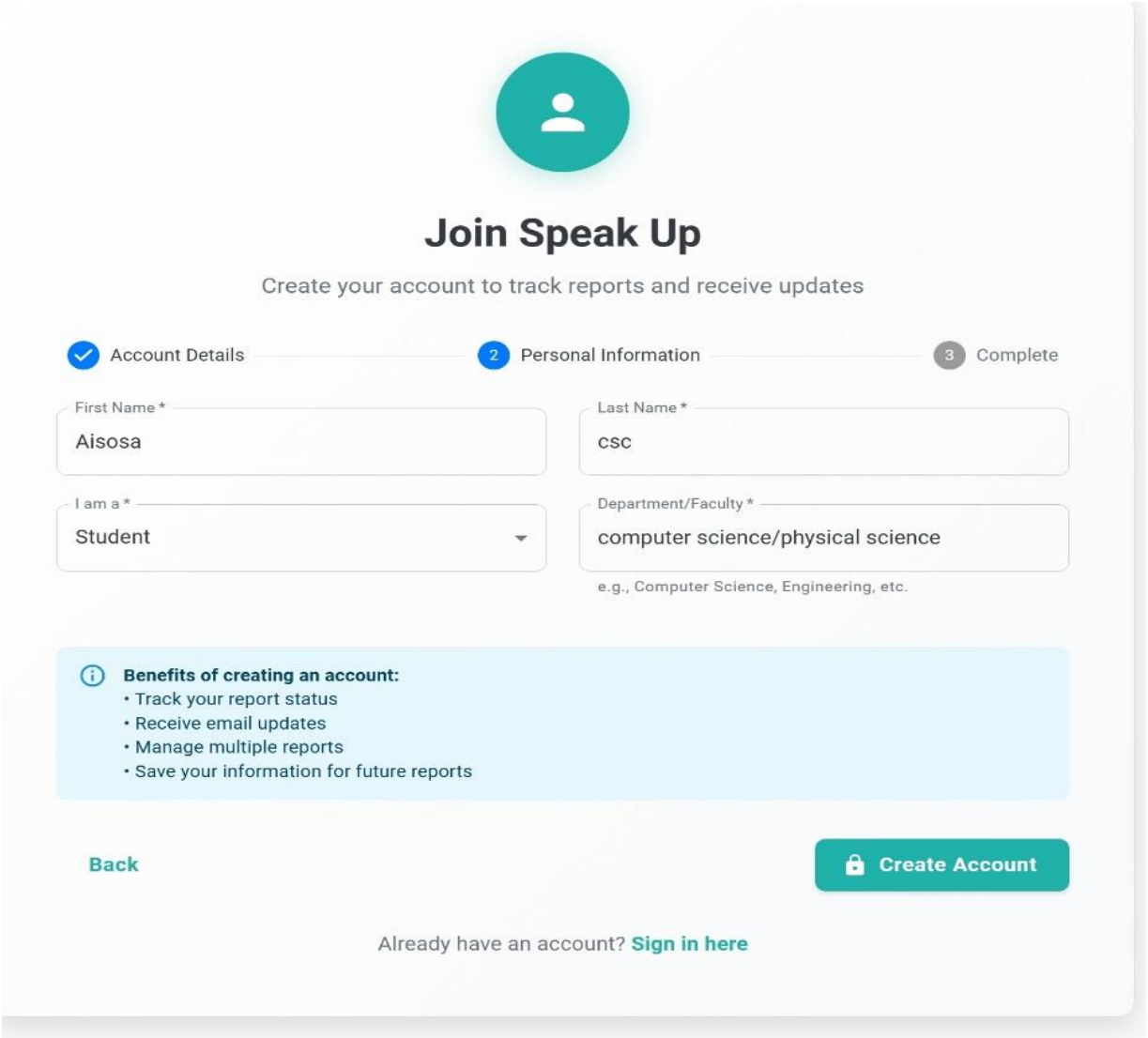


Figure 4.5 Authenticated user Registration page 2

Figure 4.4 and Figure4.5 illustrates the user registration process designed as a simple three-step workflow to facilitate secure account creation. In Step 1, users provide essential login details such as username, email, and password to establish their account credentials. Upon proceeding to Step 2, users enter personal and institutional information, including name, department, and user category. The interface also presents a brief guide outlining the benefits of account creation, such as report tracking, receiving updates, and managing multiple submissions. The structured layout and progress indicators enhance clarity, guiding users smoothly through the registration process.

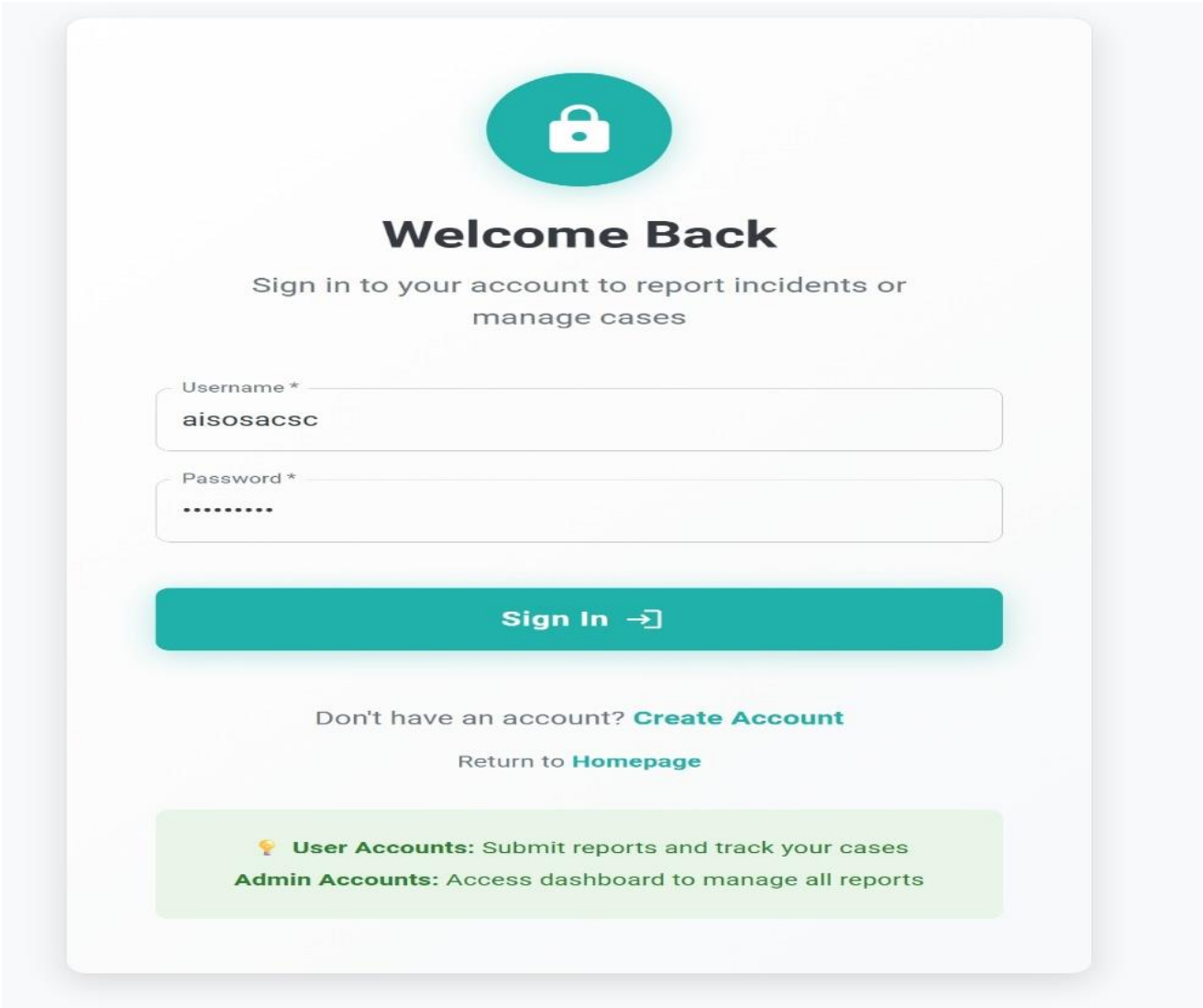


Figure 4.6 Authenticated User Login page

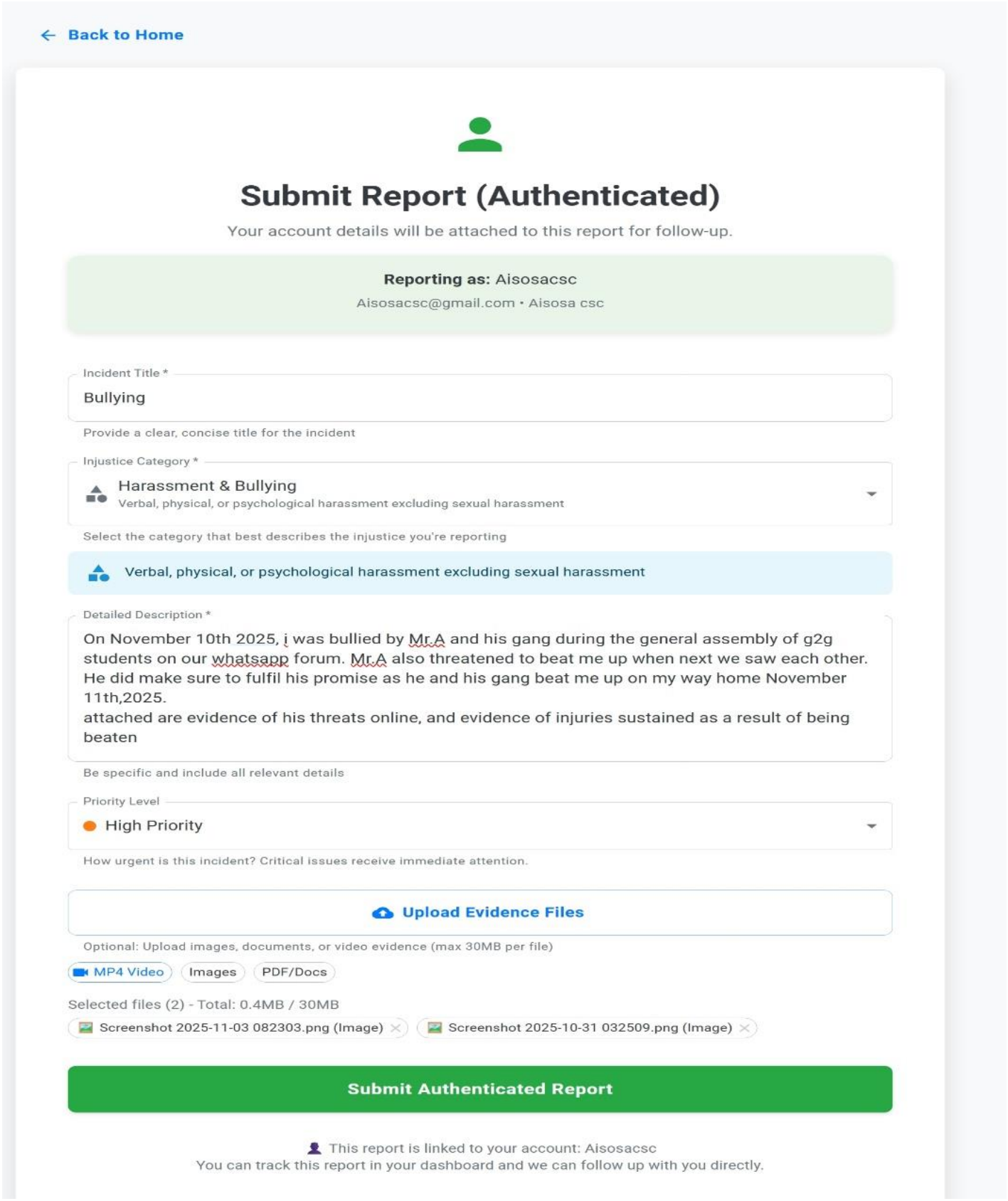


Figure 4.6 Authenticated Reporting page

Figure 4.4 illustrates the “Submit Report (Authenticated)” interface designed for registered users to file incidents under their verified accounts. The page presents a structured reporting form where users provide the incident title, injustice category, detailed description, and urgency level. It includes an upload section that supports evidence files such as videos, images, and documents to strengthen the report. At the bottom, a “Submit Authenticated Report” button enables users to finalize their submission, with a confirmation note indicating that the case will be linked to their account for tracking and follow-up within the dashboard.

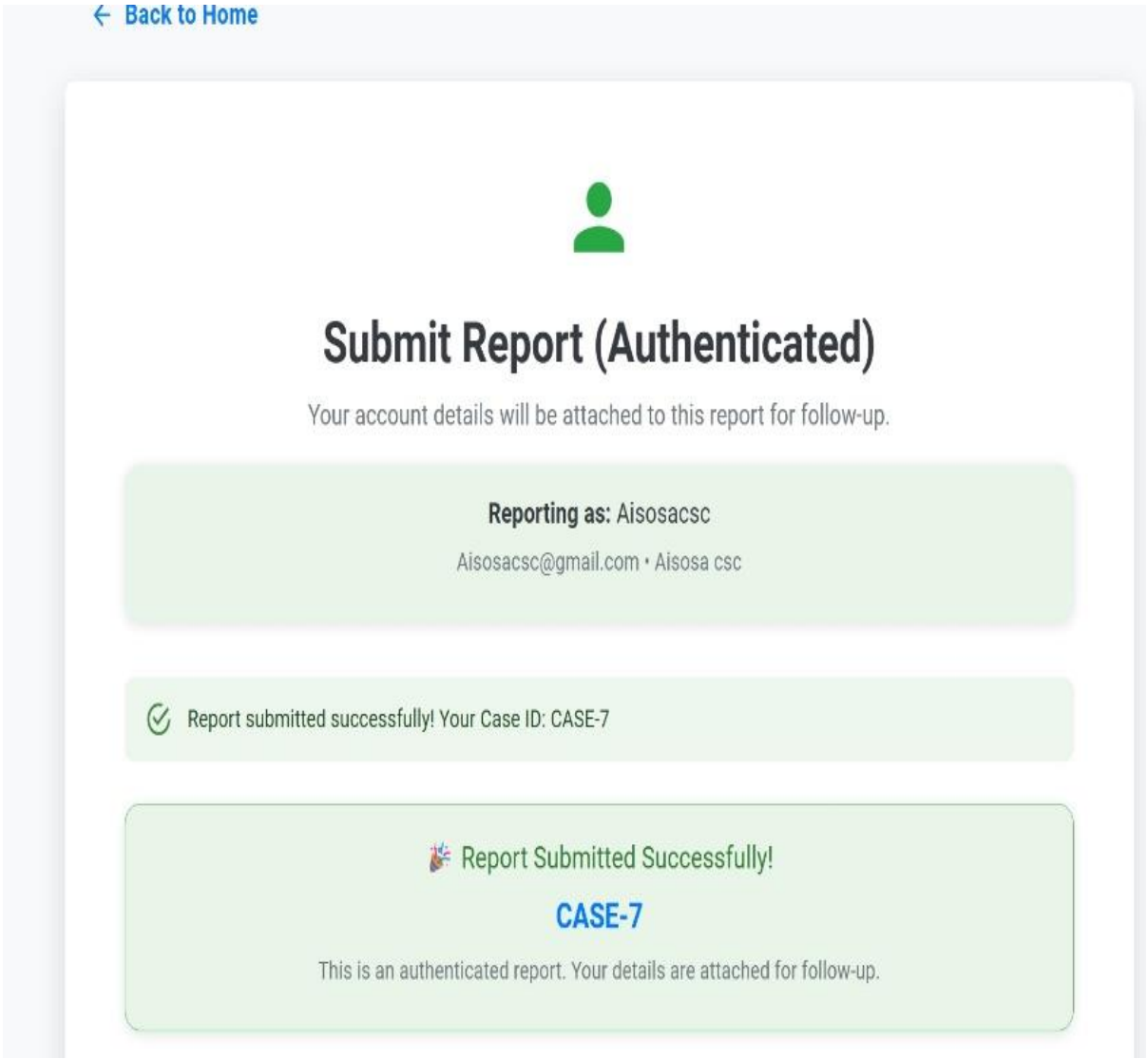


Figure 4.7 Successful Authenticated Report

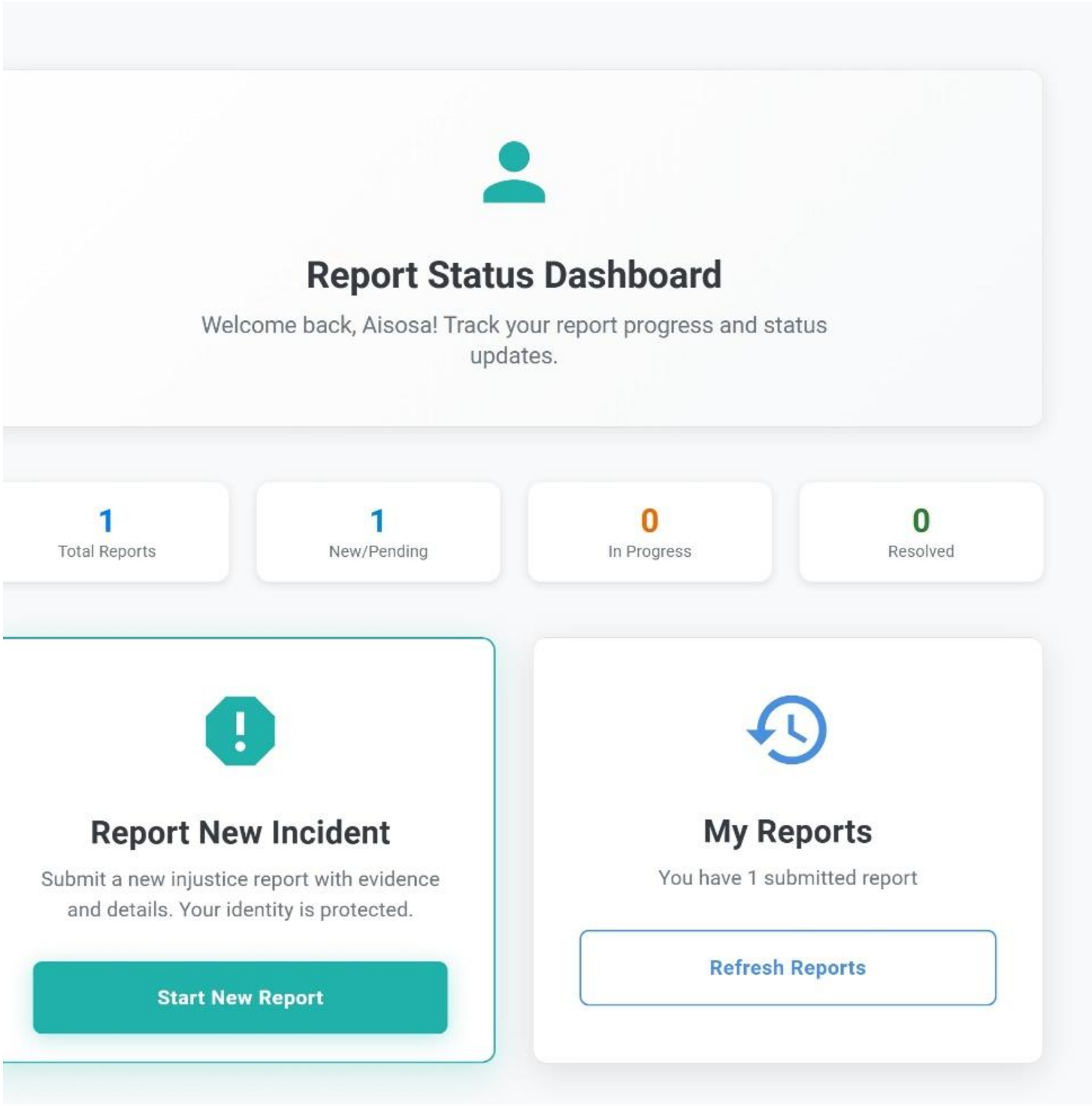


Figure 4.8 User Dashboard

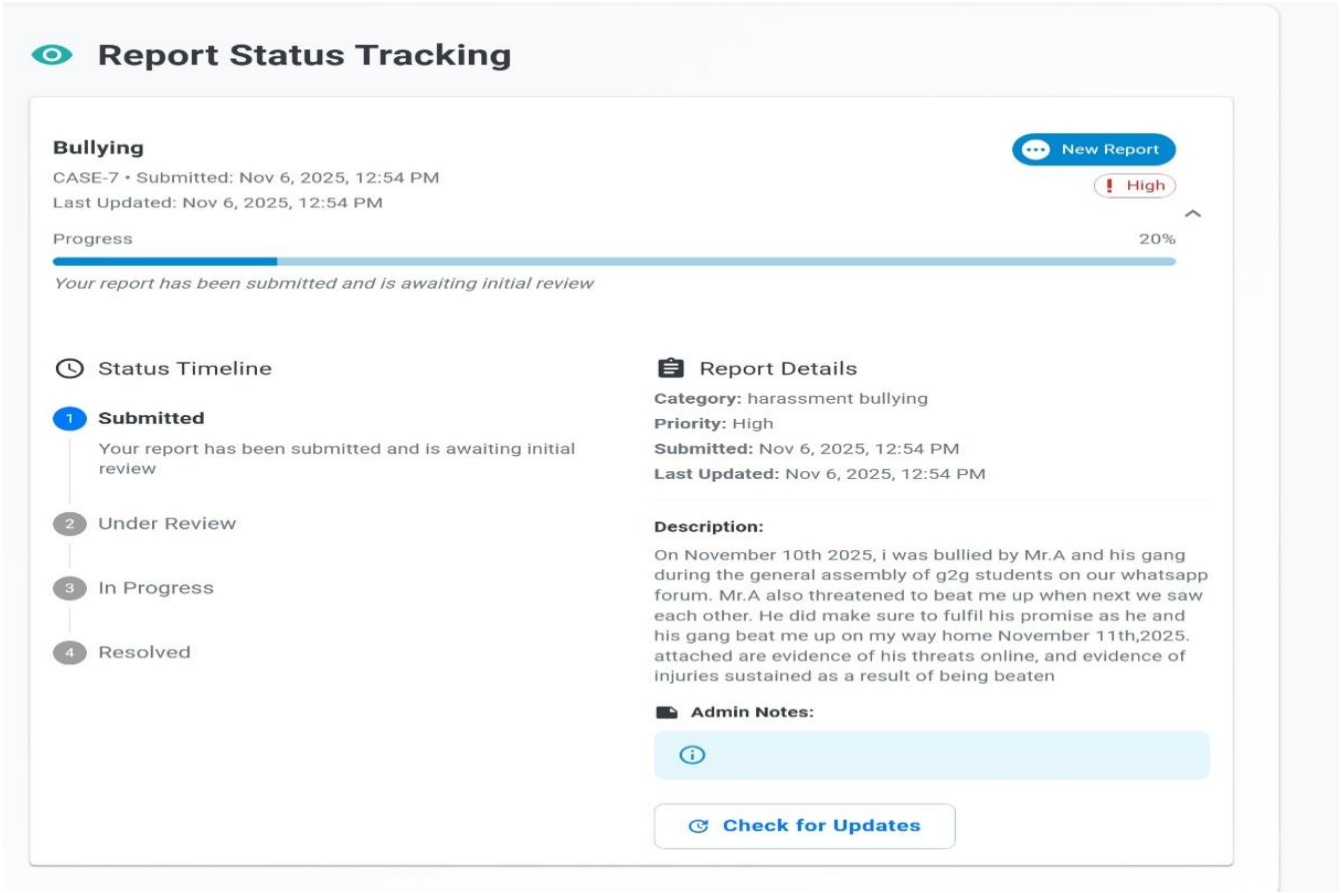


Figure:4.9 User Dashboard Report Status Tracking

Figure 4.8 and figure 4.9 presents the “Report Status Dashboard” interface of the SpeakUp web application, which provides registered users with real-time tracking of submitted cases. The dashboard header greets the user by name and displays summarized report statistics, including total reports, new or pending cases, those in progress, and resolved ones. Below, users can either submit a new report through the “Start New Report” button or view previously filed cases via “My Reports.” The status tracking section visually outlines the progress of each case using a timeline, highlighting stages such as Submitted, Under Review, In Progress, and Resolved. Detailed information on the selected report including the category, priority level, submission date, and description is displayed on the right, ensuring transparency and accountability. A tips section at the bottom provides best practices for effective reporting, encouraging users to include evidence, specify details, and track updates consistently.

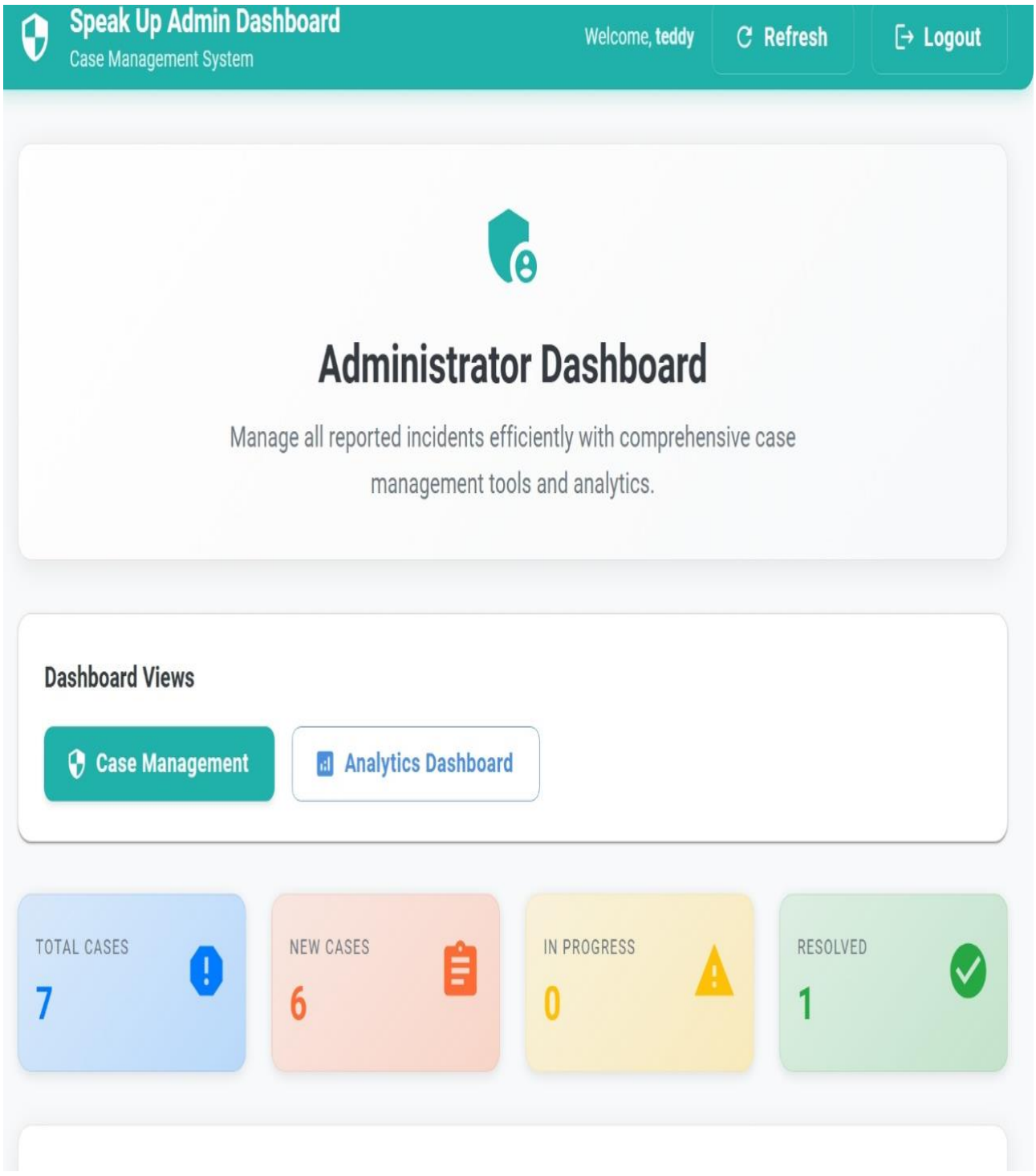
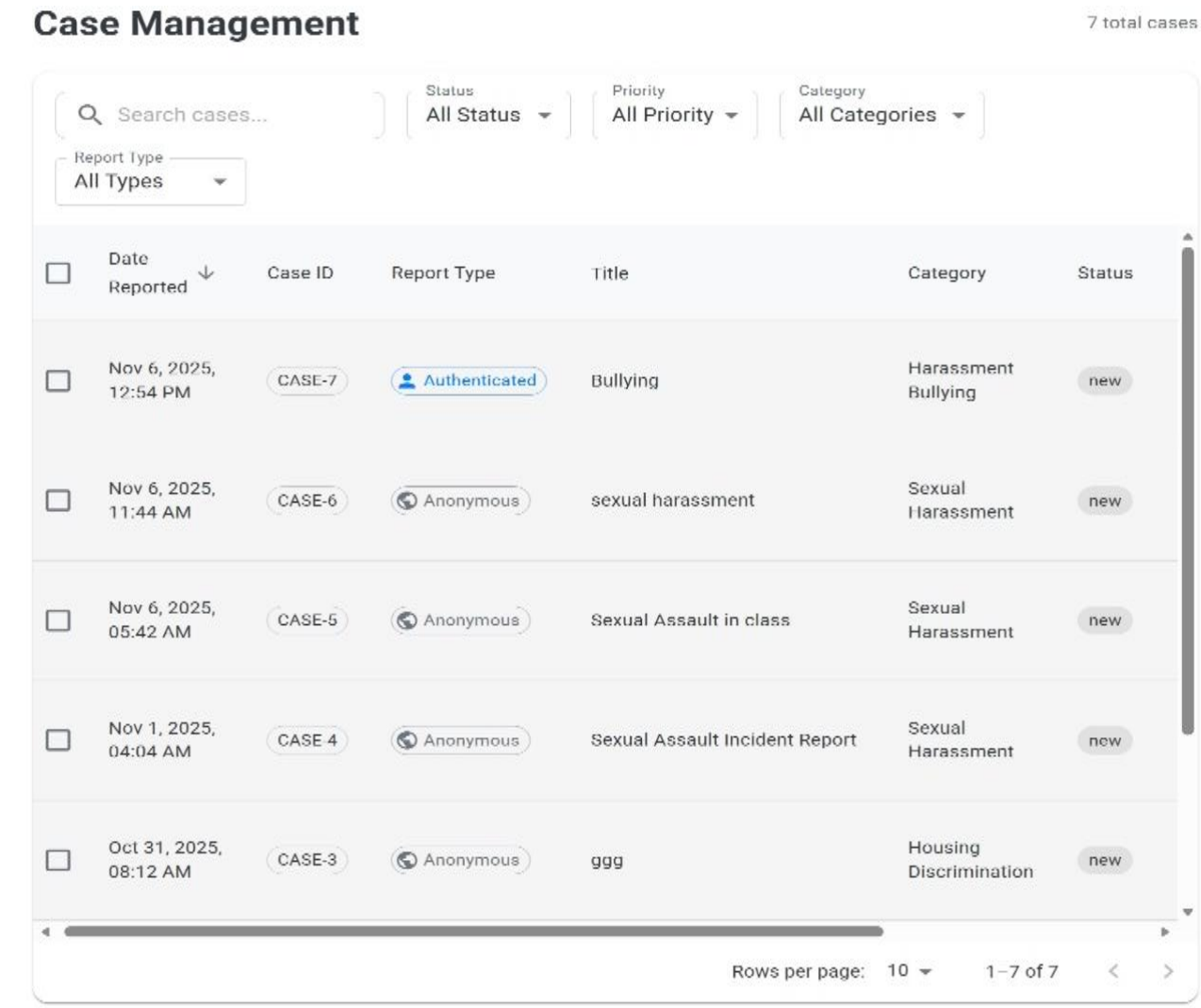


Figure 4.10 Admin dashboard

Figure 4.11 Admin dashboard: case management(A)



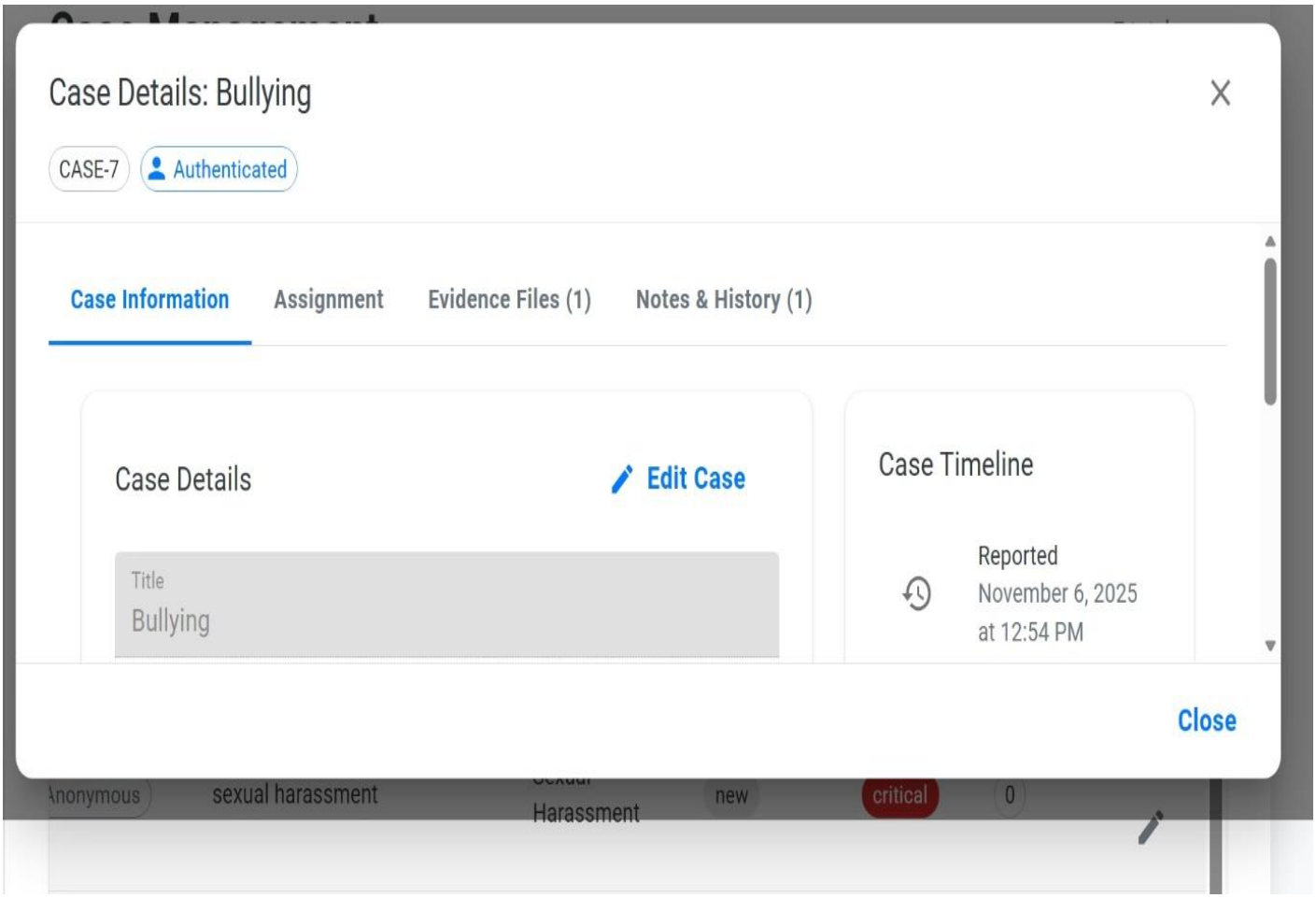


Figure 4.12 Admin dashboard: Case Management{B]

Figures 4.10 to 4.12 present the administrator's interface of the Speak Up system, illustrating the key components of case management and system oversight. The Administrator Dashboard serves as the central control panel, displaying essential system statistics such as the total number of cases, new cases, cases in progress, and resolved reports. It also includes quick-access buttons for refreshing data and logging out, ensuring efficient workflow management.

The Case Management view, accessible through the dashboard, displays a comprehensive table listing all reported cases with corresponding details such as Case ID, Report Type, Title, Category, Status, Priority, Evidence, and Actions. Administrators can easily filter and search reports using dropdown menus and search bars, enabling streamlined navigation across

multiple cases. Each report entry provides actionable icons for viewing and editing report details, promoting effective monitoring and response.

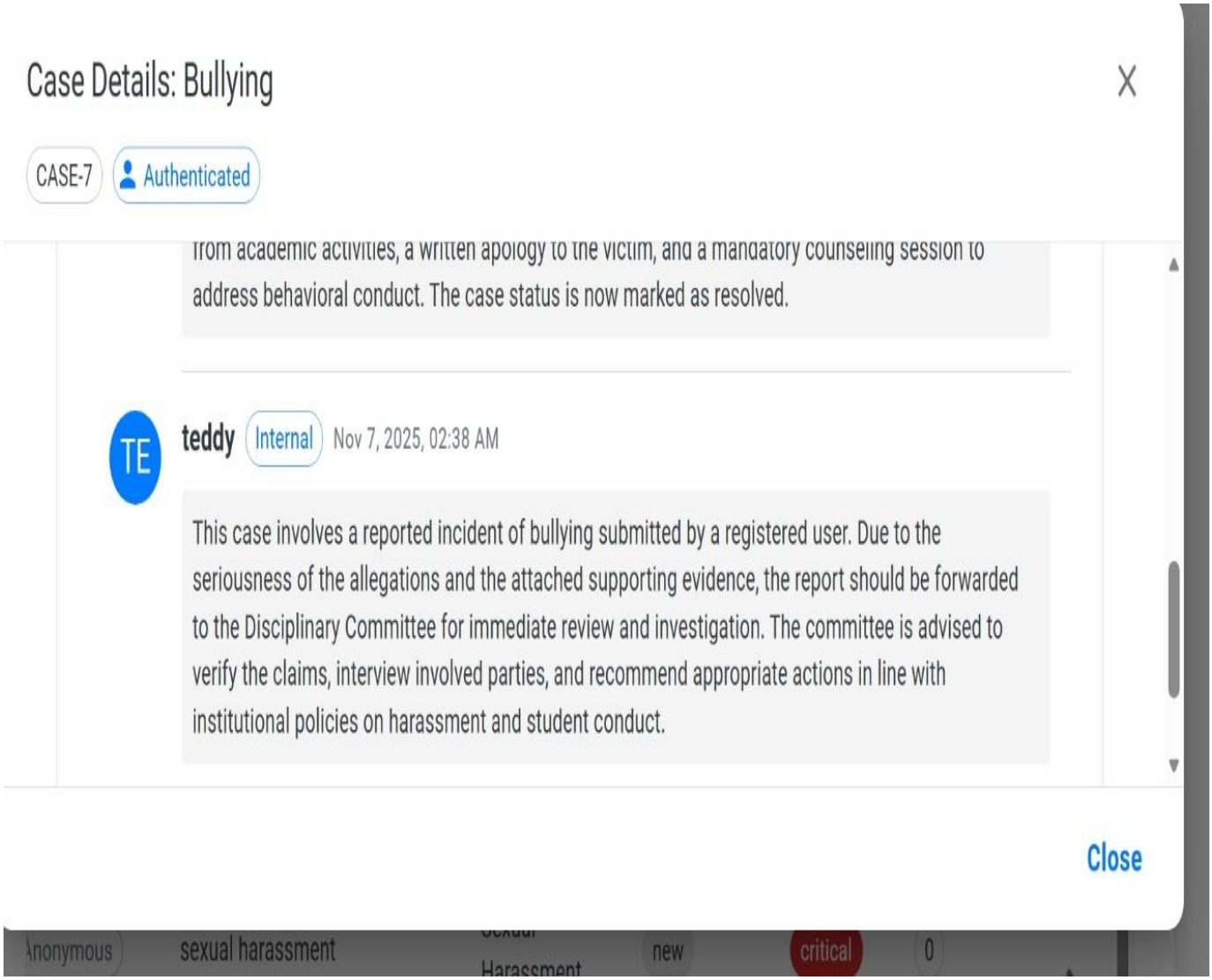


Figure 4.13 Case Details showing Admin internal notes.

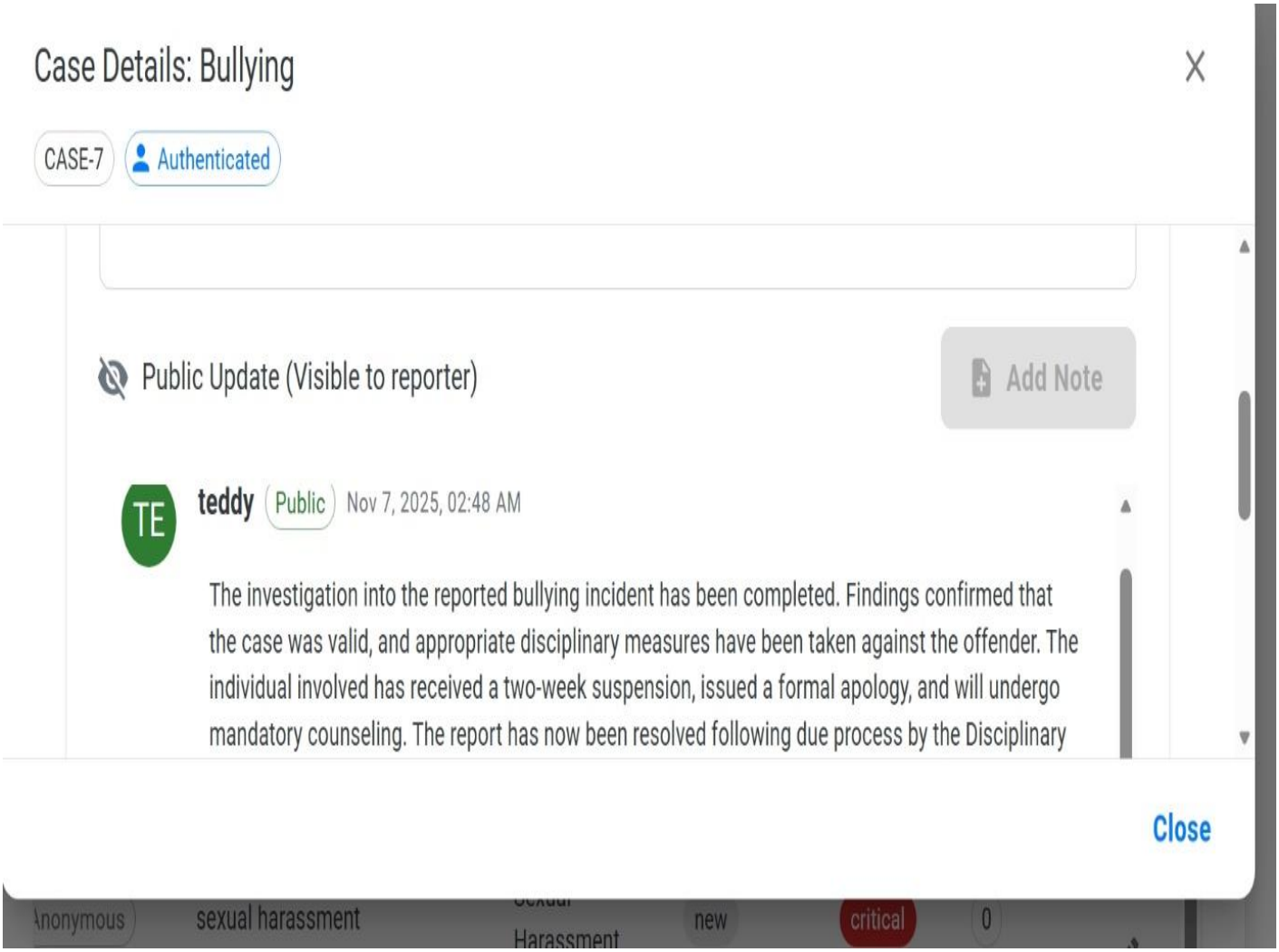


Figure 4.14 Case Details showing Admin Public notes.

Figures 4.13 to 4.15 show the Case Details modal within the administrator interface. This pop-up window presents a detailed view of a reported case, organized into multiple tabs such as Case Information, Assignment, Evidence Files, and Notes & History. The displayed section highlights how administrators can review, document, and update case progress within the system.

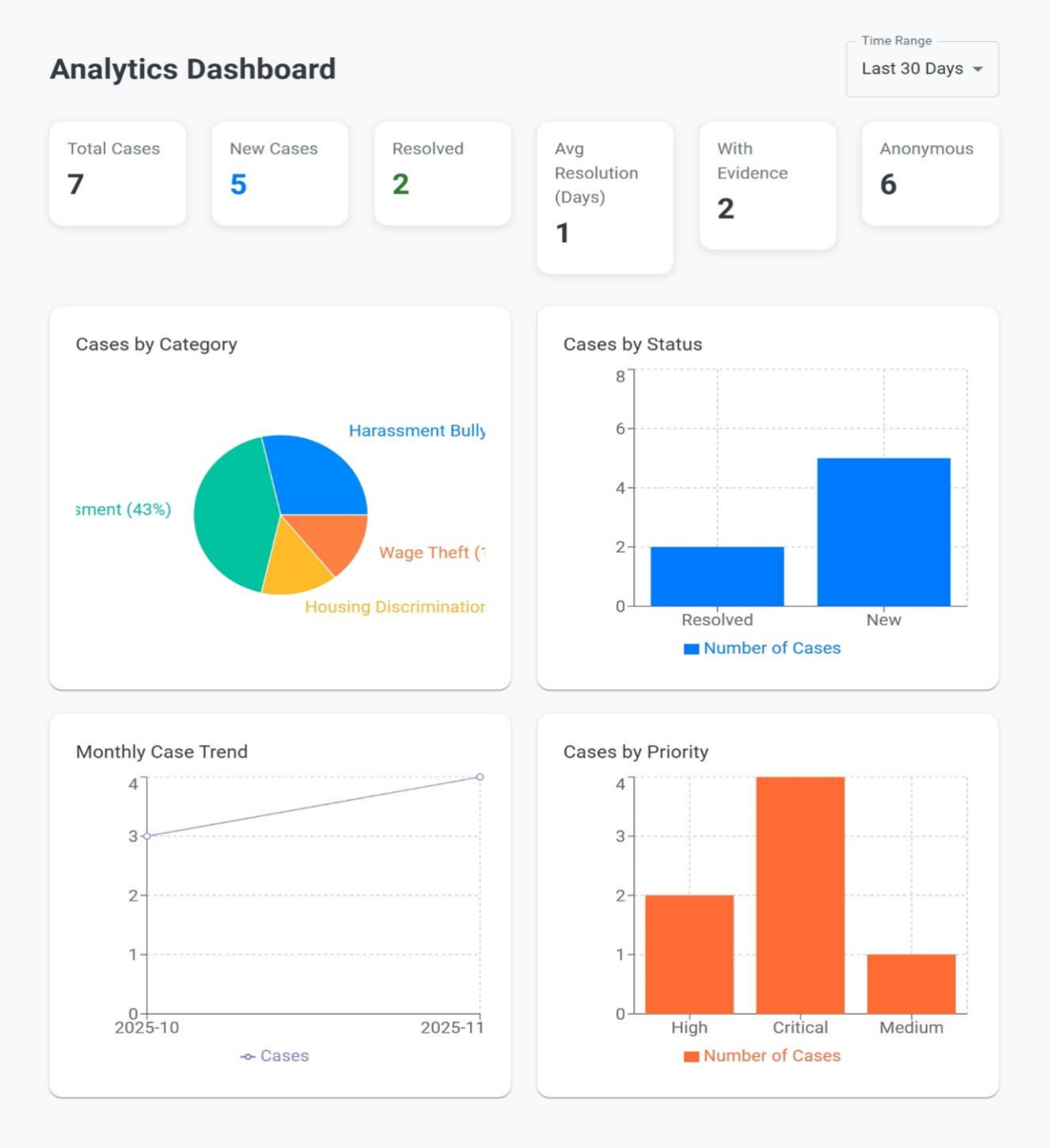


Figure 4.15 Analytic dashboard

As illustrated on Figure 4.16, The Analytics Dashboard provides a Statistical summary of handled cases within a period.

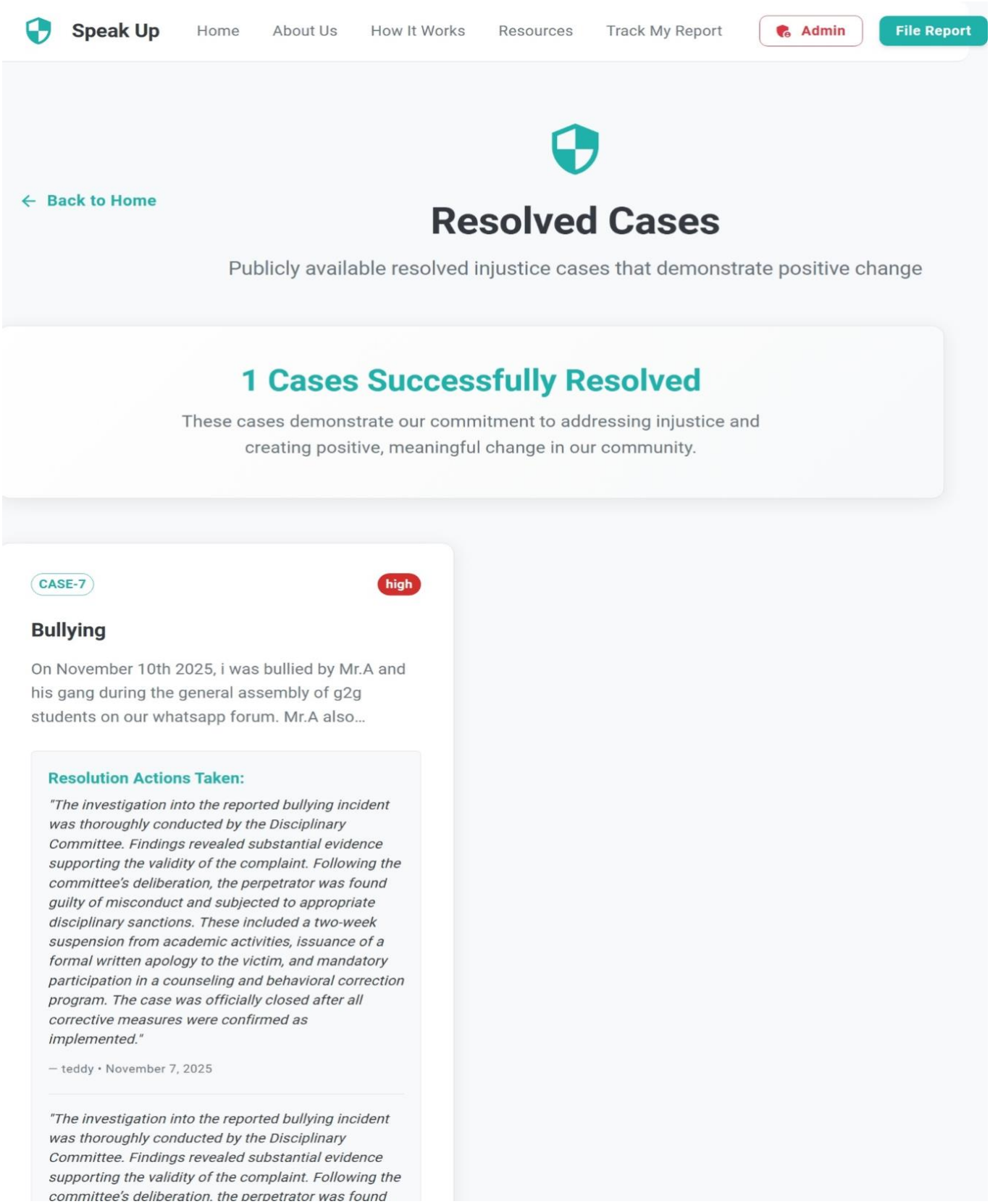


Figure 4.16 Public Displayed Resolved Case

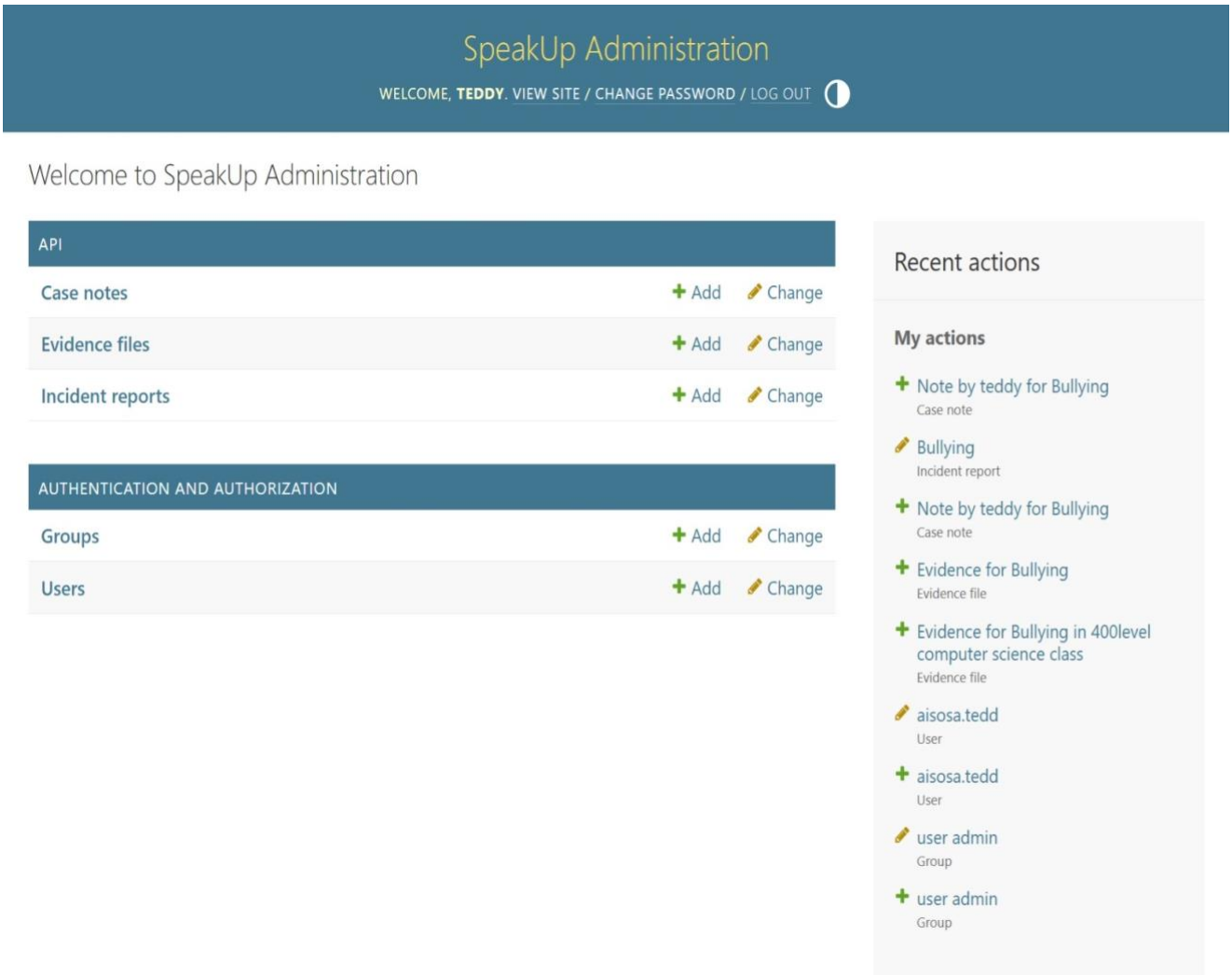


Figure 4. 17: backend Admin dashboard

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary of Findings

This research centered on the development and implementation of a web-based application known as Speak Up, designed to address issues of harassment, discrimination, and other social injustices within tertiary institutions, with a particular focus on the University of Benin. The project was inspired by the increasing need for a trustworthy and confidential reporting mechanism that allows students to report incidents without fear of stigma, retaliation, or exposure.

A structured research approach was employed using questionnaires distributed among 133 students from different faculties. The responses revealed that a significant proportion of students about 57% had personally experienced one form of injustice, while 21% had witnessed such incidents among peers. The most frequent forms of social injustice included academic favoritism (50.5%), bullying (45.9%), and financial exploitation (40.4%). It was also observed that about 82% of respondents had never reported such experiences, mainly due to a lack of confidence in existing reporting procedures, fear of victimization, and concerns over privacy.

However, a majority of respondents (over 60%) indicated that they would willingly use a digital platform if it guarantees anonymity and transparency. These insights informed the design and implementation of Speak Up, which focuses on confidentiality, user accessibility, and institutional accountability.

Technically, the Speak Up system was implemented using ReactJS for the frontend and Django (Python) for the backend, with SQLite serving as the database during development. The application supports anonymous and authenticated reporting, evidence uploads, case

management dashboards, and public access to resolved case summaries. Collectively, these features help promote a safer, more transparent, and responsive university environment.

5.2 Conclusion

The successful development of the Speak Up platform demonstrates how digital innovation can enhance justice and accountability within higher education institutions. The study established that one of the main reasons students do not report social injustices is the lack of confidentiality and confidence in existing systems. Speak Up directly addresses these barriers by providing a secure, easy-to-use, and transparent channel for reporting misconduct.

Through its dual reporting options (anonymous and authenticated), the system empowers students to share their experiences freely while allowing administrators to handle cases efficiently. The inclusion of a structured case management system, evidence tracking, and reporting analytics ensures that incidents are not only recorded but also resolved systematically. Furthermore, the public display of anonymized, resolved cases helps build trust and credibility between students and the institution.

In summary, the Speak Up application provides an effective technological solution for improving reporting processes, promoting equity, and strengthening the integrity of tertiary institutions. It stands as a model that can be replicated or expanded across other universities seeking to foster fairness, safety, and accountability.

5.3 Limitations of the Study

Although the Speak Up project achieved its primary objectives, several challenges limited its scope and implementation:

Restricted Research Coverage: Data collection was limited to students of the University of Benin. Therefore, findings may not represent the perspectives of students in other universities with different institutional cultures.

Local Hosting Constraint: The system currently operates in a local development environment, which limited public testing and user feedback during deployment.

Resource and Time Constraints: The research and development processes were limited by time and lack of financial support, preventing the integration of more advanced features such as two-factor authentication and full-scale server deployment.

Limited User Evaluation: Comprehensive usability testing with both administrators and a wider student population could not be conducted due to project time restrictions.

5.4 Recommendations

Based on the outcomes of this research and system implementation, the following recommendations are suggested:

Adoption by Institutions: Tertiary institutions should consider integrating SpeakUp or similar platforms into their administrative processes to enhance reporting and resolution of social justice concerns.

Institutional or Server Hosting: The system should be hosted on a university-managed or cloud-based server to ensure wider accessibility, improved performance, and secure data management.

Enhanced Security Features: Future versions of SpeakUp should include advanced authentication measures, encryption techniques, and automated backup systems to further safeguard user data.

Awareness and Training: Awareness campaigns should be conducted among students and staff to educate them on how to use the platform effectively and to promote confidence in the system.

Regular Updates and Maintenance: The system should be reviewed periodically to improve performance, address bugs, and include new features such as automated analytics, trend tracking, and mobile integration.

Broader Research Application: Future research should extend the system's application to multiple institutions and explore the use of artificial intelligence for automated case classification and predictive analysis.

5.5 Summary Statement

In conclusion, this study successfully designed and implemented the SpeakUp platform, a secure and user-friendly reporting system that fosters transparency, inclusivity, and fairness within tertiary institutions. The project demonstrated how digital tools can help bridge communication gaps between students and university authorities. Despite its limitations, SpeakUp provides a foundation for continued innovation in institutional accountability systems, serving as a sustainable model for promoting social justice in higher education.

REFERENCE

Adebayo, T. A., & Oyeniran, R. K. (2022). The prevalence and psychological impact of bullying among university students in Nigeria. Journal of Education and Social Research, 12(4), 134–145.

Adekunle, T., & Yusuf, A. (2021). Reporting mechanisms for social injustice in higher education institutions. Lagos University Press.

Afolabi, R., Adeyemi, L., & Ojo, M. (2021). Organizational justice and grievance reporting in Nigerian universities. African Journal of Social Sciences, 9(2), 78–92.

Akinlade, O., & Eze, P. (2022). Ensuring accessibility and confidentiality in institutional reporting systems. Nigerian Journal of Educational Management, 12(3), 45–58.

Ameh, E., Eze, C., & Oladipo, J. (2024). Sexual violence among students in Nigerian tertiary institutions: Patterns and institutional response. BMC Public Health, 24(6), 1–12.

Awojide, A. (2023). Internet usage among university students in Nigeria: Challenges in accessing and utilising online resources. African Journal of Information Science and Technology, 13(1), 16–29.

Baba, O. J. (2024). Addressing sexual harassment in Nigerian higher institutions through the feasibility of a digital anonymous feedback system. Asian Review of Social Sciences, 13(2), 18–24.

Bello, K., Adewale, F., & Chukwu, T. (2024). Gender-based harassment in Nigerian universities: Prevalence and institutional response. Nigerian Journal of Educational Research, 18(1), 12–28.

Bondestam, F., & Lundqvist, M. (2020). University awareness campaigns and reporting of misconduct: A European perspective. Routledge.

Bryman, A. (2016). Social research methods (5th ed.). Oxford University Press.

Colquitt, J., LePine, J., & Wesson, M. (2021). Organizational behavior: Improving performance and commitment in the workplace (6th ed.). McGraw-Hill Education.

Cortina, L., & Magley, V. (2021). Psychosocial support and harassment reporting: Insights from higher education. Journal of Applied Psychology in Education, 14(2), 65–80.

Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (5th ed.). SAGE Publications.

Dahiru, A. A. (2025). Enhancing public access to justice through digital tools. National Judicial Institute.

Dibie, R., & Robert, C. (2020). Social justice, governance, and higher education in Nigeria. African Journal of Public Affairs, 9(3), 101–118.

Edosomwan, E. A. (2024). Design of an online complaint management system for tertiary institutions in Nigeria. International Journal of Computer Applications, 185(6), 45–53.

Idris, A. Y. (2025). Bullying prevalence and student wellbeing in Nigerian schools: A review. Journal of Educational Psychology and Counselling, 14(1), 55–68.

Obi, A. I., & Ziprebo, T. (2024). Social injustice in tertiary educational institutions and its implications in Bayelsa State. International Journal of Library Science and Education Research, 3(8), 51–62.

Odike, M. N. (2021). Social justice in education and celebration of workforce diversity: A case in Enugu State, Nigeria. Contemporary Journal of Multidisciplinary Studies, 3(91), 21–31.

Ogunfowokan, A., Adebowale, S., & Fasina, T. (2023). Digital platforms and grievance reporting in Nigerian universities. Nigerian Journal of Educational Technology, 11(2), 77–91.

Okeke, N. (2020). Power dynamics and underreporting of misconduct in Nigerian higher education. African Review of Higher Education Studies, 8(3), 112–130.

Okeke, T. (2021). Academic corruption and student exploitation in Nigerian universities. Journal of Educational Integrity, 5(2), 20–38.

Okoro, C. E. (2023). Discrimination and its effects on student academic engagement in Nigerian universities. Journal of Educational Development Studies, 8(2), 75–84.

Oladejo, O., & Onwuzuruike, C. (2023). Challenges in basic digital reporting systems in Nigerian universities. Journal of Nigerian Educational Technology, 12(1), 10–25.

Olatunji, A., & Adebayo, R. (2022). Political exclusion and student representation in Nigerian universities. Nigerian Journal of Student Affairs, 6(1), 15–28.

Olowu, T., & Adebisi, M. (2022). Institutional bias and grievance reporting in Nigerian universities. Nigerian Educational Review, 14(2), 88–102.

Olugbenga-Bello, A. I., Adepoju, O. A., & Abodunrin, O. L. (2023). Prevalence, pattern, and perception of female sexual harassment among undergraduates in Nigeria. Nigerian Journal of Medicine, 32(4), 327–335.

Ogunfowokan, A., Adebowale, S., & Fasina, T. (2024). Reporting sexual violence in Nigerian tertiary institutions: Prevalence and system challenges. Journal of African Educational Research, 15(1), 45–60.

Shadows of Inequality. (2023). Gender, ethnicity, and power in Nigerian higher education. Lagos: Equality Press.

Saunders, M., Lewis, P., & Thornhill, A. (2019). Research methods for business students (8th ed.). Pearson Education.

Tyler, T. (2020). Procedural justice and legal compliance: Lessons from research and practice. Oxford University Press.

Waller, R., Smith, J., & Adeyemi, P. (2020). Applying social ecological theory to reporting harassment in universities. Journal of Social Issues in Education, 9(2), 55–70.

WARDC. (2022). Student experiences of harassment and reporting in Nigerian tertiary institutions. West African Research and Development Council.

Waller, R., Smith, J., & Adeyemi, P. (2020). Applying social ecological theory to reporting harassment in universities. Journal of Social Issues in Education, 9(2), 55–70.

WARDC. (2022). Student experiences of harassment and reporting in Nigerian tertiary institutions. West African Research and Development Council. https://www.wardc.org

APPENDIX

SOURCE CODE

DATABASE SCHEMA DEFINITION(MODEL.PY)

```
from django.db import models
from django.contrib.auth.models import User
import uuid
import os

def get_upload_path(instance, filename):
    ext = filename.split('.')[-1]
    filename = f"{uuid.uuid4()}.{ext}"
    return os.path.join('uploads/', filename)

class UserProfile(models.Model):
    USER_TYPE_CHOICES = [
        ('student', 'Student'),
        ('worker', 'Worker/Staff'),
    ]

    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='profile')
    department = models.CharField(max_length=100)
    user_type = models.CharField(max_length=10, choices=USER_TYPE_CHOICES)

    def __str__(self):
        return f"{self.user.username} - {self.get_user_type_display()}"

class IncidentReport(models.Model):
    STATUS_CHOICES = [
        ('new', 'New'),
        ('in_progress', 'In Progress'),
```

```
    ('resolved', 'Resolved'),
    ('closed', 'Closed'),
]

PRIORITY_CHOICES = [
    ('low', 'Low'),
    ('medium', 'Medium'),
    ('high', 'High'),
    ('critical', 'Critical'),
]

CATEGORY_CHOICES = [
    ('workplace_discrimination', 'Workplace Discrimination'),
    ('sexual_harassment', 'Sexual Harassment'),
    ('harassment_bullying', 'Harassment & Bullying'),
    ('wage_theft', 'Wage Theft & Labor Rights'),
    ('housing_discrimination', 'Housing Discrimination'),
    ('police_brutality', 'Police Brutality & Misconduct'),
    ('educational_inequality', 'Educational Inequality'),
    ('healthcare_disparity', 'Healthcare Disparity'),
    ('voter_suppression', 'Voter Suppression & Rights'),
    ('environmental_injustice', 'Environmental Injustice'),
    ('digital_discrimination', 'Digital Discrimination'),
    ('immigrant_rights', 'Immigrant Rights Violations'),
    ('disability_rights', 'Disability Rights Violations'),
    ('gender_inequality', 'Gender Inequality'),
    ('racial_injustice', 'Racial Injustice'),
    ('religious_discrimination', 'Religious Discrimination'),
    ('age_discrimination', 'Age Discrimination'),
    ('whistleblower_retaliation', 'Whistleblower Retaliation'),
    ('consumer_exploitation', 'Consumer Exploitation'),
    ('other', 'Other Injustice'),
```

```
    ]

    REPORT_TYPE_CHOICES = [
        ('anonymous', 'Anonymous Report'),
        ('authenticated', 'Authenticated Report'),
    ]

    title = models.CharField(max_length=200)
    description = models.TextField()
        category = models.CharField(max_length=50, choices=CATEGORY_CHOICES, default='other')
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='new')
        priority = models.CharField(max_length=20, choices=PRIORITY_CHOICES, default='medium')
     report_type = models.CharField(max_length=20, choices=REPORT_TYPE_CHOICES, default='anonymous')
    created_by = models.ForeignKey(
        User,
        on_delete=models.SET_NULL,
        null=True,
        blank=True,
        related_name='incident_reports'
    )
    assigned_to = models.ForeignKey(
        User,
        on_delete=models.SET_NULL,
        null=True,
        blank=True,
        related_name='assigned_cases'
    )
    is_public = models.BooleanField(
        default=False,
        help_text="Make this case publicly visible when resolved"
```

```
    )
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.title

    class Meta:
        ordering = ['-created_at']

class EvidenceFile(models.Model):
    report = models.ForeignKey(IncidentReport, on_delete=models.CASCADE, related_name='evidence_files')
    file = models.FileField(upload_to=get_upload_path)
    uploaded_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Evidence for {self.report.title}"

    def filename(self):
        return os.path.basename(self.file.name)

    class Meta:
        ordering = ['-uploaded_at']

class CaseNote(models.Model):
    report = models.ForeignKey(IncidentReport, on_delete=models.CASCADE, related_name='notes')
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    content = models.TextField()
    is_internal = models.BooleanField(default=True)
    is_public = models.BooleanField(
        default=False,
```

```
        help_text="Make this note visible on public case pages"
    )
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f"Note by {self.author.username} for {self.report.title}"

    class Meta:
        ordering = ['-created_at']
```

CORE APPLICATION LOGIC

```
from rest_framework import generics, permissions, status, viewsets
from rest_framework.response import Response
from rest_framework.decorators import api_view, permission_classes, action
from django.contrib.auth.models import User
from .models import IncidentReport, EvidenceFile, CaseNote
from .serializers import (
    IncidentReportSerializer, IncidentReportCreateSerializer,
    EvidenceFileSerializer, UserRegistrationSerializer, UserSerializer,
    AnonymousReportSerializer, AuthenticatedReportSerializer,
    CaseNoteSerializer, CaseNoteCreateSerializer
)

class IsOwnerOrAdmin(permissions.BasePermission):
    """Custom permission to only allow owners or admins to access objects."""

    def has_object_permission(self, request, view, obj):
        # Admin users can do anything
        if request.user.is_staff:
            return True
        # Regular users can only access their own reports
```

```
        return obj.created_by == request.user

class IsAdminUser(permissions.BasePermission):
    """Permission to only allow admin users."""

    def has_permission(self, request, view):
        return request.user and request.user.is_staff

class EvidenceFileViewSet(viewsets.ModelViewSet):
    permission_classes = [permissions.IsAuthenticated]
    serializer_class = EvidenceFileSerializer

    def get_queryset(self):
        queryset = EvidenceFile.objects.all()
        report_id = self.request.query_params.get('report')
        if report_id:
            queryset = queryset.filter(report_id=report_id)
        return queryset

    def perform_create(self, serializer):
        report_id = self.request.data.get('report')
        if not report_id:
            from rest_framework.exceptions import ValidationError
            raise ValidationError("Report ID is required")

        try:
            if self.request.user.is_staff:
                # Admin can add evidence to any report
                report = IncidentReport.objects.get(id=report_id)
            else:
                # Regular users can only add evidence to their own reports
                report = IncidentReport.objects.get(
```

```
                id=report_id,
                created_by=self.request.user
            )
            serializer.save(report=report)
        except IncidentReport.DoesNotExist:
            from rest_framework.exceptions import PermissionDenied
            raise PermissionDenied("You don't have permission to add evidence to this report.")

class CaseNoteViewSet(viewsets.ModelViewSet):
    permission_classes = [permissions.IsAuthenticated]

    def get_serializer_class(self):
        if self.request.method == 'POST':
            return CaseNoteCreateSerializer
        return CaseNoteSerializer

    def get_queryset(self):
        queryset = CaseNote.objects.all()
        report_id = self.request.query_params.get('report')
        if report_id:
            queryset = queryset.filter(report_id=report_id)

        # Non-admin users can only see notes for their own reports
        if not self.request.user.is_staff:
            queryset = queryset.filter(report__created_by=self.request.user)

        return queryset.order_by('-created_at')

    def perform_create(self, serializer):
        report_id = self.request.data.get('report')
        if not report_id:
            from rest_framework.exceptions import ValidationError
```

```
            raise ValidationError("Report ID is required")

        try:
            if self.request.user.is_staff:
                # Admin can add notes to any report
                report = IncidentReport.objects.get(id=report_id)
            else:
                # Regular users can only add notes to their own reports
                report = IncidentReport.objects.get(
                    id=report_id,
                    created_by=self.request.user
                )

            # Pass report to serializer context
            serializer.save(
                author=self.request.user,
                report=report
            )
        except IncidentReport.DoesNotExist:
            from rest_framework.exceptions import PermissionDenied
            raise PermissionDenied("You don't have permission to add notes to this report.")

class IncidentReportViewSet(viewsets.ModelViewSet):
    permission_classes = [permissions.IsAuthenticated]

    def get_serializer_class(self):
        if self.request.method == 'POST':
            return IncidentReportCreateSerializer
        return IncidentReportSerializer

    def get_queryset(self):
        user = self.request.user
```

```
    if user.is_staff:
        return IncidentReport.objects.all()
    return IncidentReport.objects.filter(created_by=user)

def perform_create(self, serializer):
    if self.request.user.is_authenticated:
        serializer.save(
            created_by=self.request.user,
            report_type='authenticated'
        )
    else:
        serializer.save(
            created_by=None,
            report_type='anonymous'
        )

# Custom action for evidence files
@action(detail=True, methods=['get'])
def evidence(self, request, pk=None):
    report = self.get_object()
    evidence_files = report.evidence_files.all()
    serializer = EvidenceFileSerializer(evidence_files, many=True)
    return Response(serializer.data)

# Custom action for case notes
@action(detail=True, methods=['get'])
def notes(self, request, pk=None):
    report = self.get_object()
    notes = report.notes.all()
    serializer = CaseNoteSerializer(notes, many=True)
    return Response(serializer.data)
```

```
# Anonymous Report Creation (Public - no authentication required)
class AnonymousReportCreate(generics.CreateAPIView):
    permission_classes = [permissions.AllowAny]
    serializer_class = AnonymousReportSerializer

    def perform_create(self, serializer):
        serializer.save(
            report_type='anonymous',
            status='new',
            created_by=None
        )

# Authenticated Report Creation (Logged-in users only)
class AuthenticatedReportCreate(generics.CreateAPIView):
    permission_classes = [permissions.IsAuthenticated]
    serializer_class = AuthenticatedReportSerializer

    def perform_create(self, serializer):
        serializer.save(
            report_type='authenticated',
            status='new',
            created_by=self.request.user
        )

class IncidentReportListCreate(generics.ListCreateAPIView):
    def get_permissions(self):
        if self.request.method == 'POST':
            return [permissions.AllowAny()]  # Anyone can submit (backward compatibility)
        return [permissions.IsAuthenticated()]  # Only auth users can view

    def get_serializer_class(self):
        if self.request.method == 'POST':
```

```
            return IncidentReportCreateSerializer
        return IncidentReportSerializer

    def get_queryset(self):
        user = self.request.user
        if user.is_staff:
            return IncidentReport.objects.all()
        return IncidentReport.objects.filter(created_by=user)

    def perform_create(self, serializer):
        if self.request.user.is_authenticated:
            serializer.save(
                created_by=self.request.user,
                report_type='authenticated'
            )
        else:
            serializer.save(
                created_by=None,
                report_type='anonymous'
            )

class IncidentReportDetail(generics.RetrieveUpdateDestroyAPIView):
    permission_classes = [permissions.IsAuthenticated, IsOwnerOrAdmin]
    serializer_class = IncidentReportSerializer

    def get_queryset(self):
        user = self.request.user
        if user.is_staff:
            return IncidentReport.objects.all()
        return IncidentReport.objects.filter(created_by=user)

class EvidenceFileCreate(generics.CreateAPIView):
```

```
    permission_classes = [permissions.IsAuthenticated]
    serializer_class = EvidenceFileSerializer

    def perform_create(self, serializer):
        report_id = self.kwargs['report_id']
        try:
            if self.request.user.is_staff:
                # Admin can add evidence to any report
                report = IncidentReport.objects.get(id=report_id)
            else:
                # Regular users can only add evidence to their own reports
                report = IncidentReport.objects.get(
                    id=report_id,
                    created_by=self.request.user
                )
            serializer.save(report=report)
        except IncidentReport.DoesNotExist:
            from rest_framework.exceptions import PermissionDenied
            raise PermissionDenied("You don't have permission to add evidence to this report.")


@api_view(['POST'])
@permission_classes([permissions.AllowAny])
def register_user(request):
    """Register a new regular user (non-admin)"""
    serializer = UserRegistrationSerializer(data=request.data)
    if serializer.is_valid():
        user = serializer.save()
        # Ensure new users are NOT staff/admin by default
        user.is_staff = False
        user.is_superuser = False
        user.save()
        return Response({
```

```
            'user': UserSerializer(user).data,
            'message': 'Regular user created successfully'
        }, status=status.HTTP_201_CREATED)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


@api_view(['GET'])
def get_user_profile(request):
    """Get current user profile"""
    if not request.user.is_authenticated:
        return Response({'error': 'Authentication required'}, status=status.HTTP_401_UNAUTHORIZED)
    serializer = UserSerializer(request.user)
    return Response(serializer.data)


@api_view(['GET'])
def get_my_reports(request):
    """Endpoint for users to see ONLY their reports"""
    if not request.user.is_authenticated:
        return Response({'error': 'Authentication required'}, status=status.HTTP_401_UNAUTHORIZED)

    if request.user.is_staff:
        reports = IncidentReport.objects.all()
    else:
        reports = IncidentReport.objects.filter(created_by=request.user)

    serializer = IncidentReportSerializer(reports, many=True)
    return Response(serializer.data)


@api_view(['GET'])
@permission_classes([permissions.IsAuthenticated, IsAdminUser])
def get_staff_users(request):
    """Get all staff users for case assignment"""
```

```
    staff_users = User.objects.filter(is_staff=True)
    serializer = UserSerializer(staff_users, many=True)
    return Response(serializer.data)


@api_view(['GET'])
@permission_classes([permissions.IsAuthenticated, IsAdminUser])
def admin_dashboard(request):
    """Admin-only dashboard statistics"""
    if not request.user.is_staff:
        return Response({'error': 'Admin access required'}, status=status.HTTP_403_FORBIDDEN)

    total_reports = IncidentReport.objects.count()
    open_reports = IncidentReport.objects.filter(status='new').count()
    in_progress_reports = IncidentReport.objects.filter(status='in_progress').count()
    resolved_reports = IncidentReport.objects.filter(status='resolved').count()
    closed_reports = IncidentReport.objects.filter(status='closed').count()

    # Add report type statistics
    anonymous_reports = IncidentReport.objects.filter(report_type='anonymous').count()
    authenticated_reports = IncidentReport.objects.filter(report_type='authenticated').count()

    return Response({
        'total_reports': total_reports,
        'open_reports': open_reports,
        'in_progress_reports': in_progress_reports,
        'resolved_reports': resolved_reports,
        'closed_reports': closed_reports,
        'anonymous_reports': anonymous_reports,
        'authenticated_reports': authenticated_reports,
        'total_users': User.objects.count(),
    })
```

```
@api_view(['POST'])
@permission_classes([permissions.IsAuthenticated])
def test_user_assignment(request):
    """Test endpoint to verify user assignment is working"""
    from .serializers import IncidentReportCreateSerializer
    serializer = IncidentReportCreateSerializer(data=request.data)
    if serializer.is_valid():
        # This should always assign the logged-in user
        report = serializer.save(
            created_by=request.user,
            report_type='authenticated'
        )
        return Response({
            'message': 'Report created successfully',
            'report_id': report.id,
            'created_by': request.user.username,
            'report_type': report.report_type,
            'user_authenticated': request.user.is_authenticated
        }, status=status.HTTP_201_CREATED)
    return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

# Get reports by type for admin analysis
@api_view(['GET'])
@permission_classes([permissions.IsAuthenticated, IsAdminUser])
def get_reports_by_type(request):
    """Get reports filtered by report type (admin only)"""
    report_type = request.GET.get('type', 'all')

    if report_type == 'anonymous':
        reports = IncidentReport.objects.filter(report_type='anonymous')
    elif report_type == 'authenticated':
```

```
        reports = IncidentReport.objects.filter(report_type='authenticated')
    else:
        reports = IncidentReport.objects.all()

    serializer = IncidentReportSerializer(reports, many=True)
    return Response(serializer.data)

@api_view(['GET'])
@permission_classes([permissions.AllowAny])
def get_public_resolved_cases(request):
    """Get publicly available resolved cases with their public notes"""
    # Get public resolved cases
    public_cases = IncidentReport.objects.filter(
        status='resolved',
        is_public=True
    ).order_by('-updated_at')

    # Create a custom response that includes cases with their public notes
    cases_data = []
    for case in public_cases:
        case_data = IncidentReportSerializer(case).data

        # Get public notes for this case
        public_notes = case.notes.filter(is_public=True).order_by('-created_at')
        public_notes_data = CaseNoteSerializer(public_notes, many=True).data

        # Add public notes to case data
        case_data['public_notes'] = public_notes_data
        cases_data.append(case_data)

    return Response(cases_data)
```

API URL ROUTING

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView
from . import views

# Create a SINGLE router instance
router = DefaultRouter()
router.register(r'reports', views.IncidentReportViewSet, basename='report')
router.register(r'evidence', views.EvidenceFileViewSet, basename='evidence')
router.register(r'notes', views.CaseNoteViewSet, basename='note')

urlpatterns = [
    # Router URLs (includes /api/evidence/ and /api/notes/)
    path('', include(router.urls)),

    # Authentication (JWT)
    path('token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
    path('register/', views.register_user, name='register'),
    path('profile/', views.get_user_profile, name='profile'),

    # User-specific endpoints
    path('my-reports/', views.get_my_reports, name='my-reports'),

    # Admin-only endpoints
    path('admin/dashboard/', views.admin_dashboard, name='admin-dashboard'),
    path('admin/reports-by-type/', views.get_reports_by_type, name='reports-by-type'),
    path('admin/staff-users/', views.get_staff_users, name='staff-users'),

    # Incident Reports - SEPARATE ENDPOINTS FOR DIFFERENT REPORT TYPES
    path('reports/anonymous/', views.AnonymousReportCreate.as_view(), name='anonymous-
report'),
```

```
    path('reports/authenticated/', views.AuthenticatedReportCreate.as_view(), name='authenticated-report'),

    # Legacy endpoints (maintained for backward compatibility)
    path('reports/', views.IncidentReportListCreate.as_view(), name='report-list-create'),
    path('reports/<int:pk>/', views.IncidentReportDetail.as_view(), name='report-detail'),

    # Evidence Files (legacy)
    path('reports/<int:report_id>/evidence/', views.EvidenceFileCreate.as_view(), name='evidence-create'),

    # Test endpoint
    path('test-assignment/', views.test_user_assignment, name='test-assignment'),
    path('reports/public/resolved/', views.get_public_resolved_cases, name='public-resolved-cases'),
]
```