UNIVERSITY OF DAR ES SALAAM COMPUTING CENTER (UCC)

FINAL YEAR PROJECT REPORT

TITLE: VISITOR INFORMATION SYSTEM (VIS)

CASE STUDY: TANZANIA BUREAU OF STANDARDS

(TBS)

BY:

REG.NO.	NAME
01-0421-02-54087	DOGAN ROBERT STEPHANO
01-0421-02-52517	VALENTINO JAMES MLANGWA
01-0421-02-53718	NAJIM ZUBERI AHMADI
01-0421-02-54151	GRACE PAUL NYINGI
01-0421-02-53935	JACOB CHRISTOPHER MIHAMBO

A Project Report submitted in partial fulfilment of the requirements for the Diploma in Computing & Information Technology & Business and Information Technology of University of Dar es Salaam Computing Centre (UCC)

April 2023

ABSTRACT

The study aimed to improve the current visitor information system, which is mainly based on manual recording of visitor information and lacks a comprehensive database. This results in difficulty with accountability and time-consuming processes. Additionally, visitors may not provide complete information due to the paper and pen method of recording, leading to queues during information entry.

The study was conducted at the Tanzania Bureau of Standards located in Ubungo, Dar es Salaam, where we analyzed the current system and identified areas for improvement. Overall, the study aimed to improve the Visitor Information system and make it more effective and efficient

Based on our findings, we recommend the implementation of a user IDS card verification that are valid for visitors to gain access to the premises or organization. This will enhance security and reduce the risk of unauthorized entry. Additionally, we would like to recommend the implementation of the visitor feedback rating feature that will allow visitors to provide valuable feedback to improve the quality of services.

Therefore, the implementation of the new system at Tanzania Bureau of Standards will increase the efficiency of the company and will reduce the duplication of data since information are managed through the database system.

CERTIFICATION

The undersigned certify that he has read and hereby recommend for acceptance by the University of Dar es Salaam Computing Centre a Final Year Project Report entitled Visitor Information System (VIS): The Case Study of TANZANIA BUREAU OF STANDARDS (TBS) in Dar es Salaam, in fulfillment of the requirements for the Diploma in Computing & Information Technology of University of Dar es Salaam Computing Centre

…….

MacDonald Machange

(Supervisor)

Date: ----------

ACKNOWLEDGEMENTS

We express our highest gratitude and appreciation to the following individuals, parties, and organizations for their outstanding support, commitment, and timeless contribution to the success of this final year project. Firstly, we extend our deepest appreciation to our esteemed supervisor, Mr. MacDonald Machange, for providing us with valuable information and the right guidance and advice to complete this project. We also want to acknowledge the respected head of ICT at Tanzania Bureau of Standards, Mr. JABIR, for allowing us to undergo this study and providing us with the necessary facilities. In addition, we would like to express our appreciation to the ICT members, especially Mr. Abdullatif, for their crucial role in this project. Special thanks also go to Mr. Prudence Ongera for his support, guidance, teachings, and suggestions that helped improve our knowledge and attitude towards accomplishing the goals of this course. We would also like to thank Tanzania Bureau of Standards organization and our friends who helped us to assemble the parts and gave suggestions about the project. Lastly, we express our gratitude to ChatGpt, an AI platform that helped us search for the world's information, including webpages, and more. Its special features were of great help in finding exactly what we were looking for on the world wide web.

TABLE OF CONTENTS

ABSTRACT ii
CERTIFICATION iii
ACKNOWLEDGEMENTS iv
LIST OF TABLES vi
LIST OF FIGURES vi
LIST OF ABBREVIATIONS vii
CHAPTER ONE 1
INTRODUCTION 1
1.1 Background of the project 1
1.2 Background of the Organization 1
1.3 Statement of the problems 2
1.4.1 General Objectives 2
1.4.2 Specific Objectives 2
CHAPTER TWO 4
REQUIREMENTS ANALYSIS 4
2.1 Overview of the existing System 4
2.1.1 Activities of the system 4
2.1.2 Problem of existing system 4
2.2 Over view of the proposed system 4
2.2.1 Functional requirements 6
2.2.1.4 System administration 7
2.2.2 Non-functional requirements 8
2.2.3 System requirements (Hardware and software requirements) 9
CHAPTER THREE 11
USE CASES 11
3.1 Use case Model 11
3.1.1 Actors' specification 11
3.1.2 Use Case Diagram 13
3.1.3 Use Case Description 13
Actor Admin 14
Reporting User 14
Actor User 16

CHAPTER FOUR 17
SYSTEM DESIGN 17
4.1 Design Goals 17
4.2 System Decomposition 17
4.3 Persistence data management 19
4.4 Access control and security 21
4.5 User Interface design 22
Figure 7 VIS admin Dashboard 24
Figure 8 VIS User Interface 24
CHAPTER FIVE 30
CONCLUSION, EVALUATION AND FURTHER WORK 30
5.1 Conclusion 30
5.2 What we have achieved 30
5.3 Further Work 31
References 31
Appendices 32
Appendices 1:1 Login Source Code 32
Appendices 1:2 Admin Dashboard Source Codes 34
Appendices 1:3 Users Dashboard Source Codes 35

LIST OF TABLES

Table 1 Admin use case. 14
Table 2 Reporting User Use case 15
Table 3 User use case. 16

LIST OF FIGURES

Figure 1 Use Case Diagram for Visitor Information System 13
Figure 2 Functional Decomposition 18
Figure 3 Context Diagram 18
Figure 4 VIS DFD LEVEL 1 19
Figure 5 VIS Entity Relational Diagram 20
Figure 6 login form for vis 23
Figure 7 VIS admin Dashboard 24
Figure 8 VIS User Interface 24
Figure 9 Vis Users Management View 25

Figure 10 Interface For Visitor Registration 25
Figure 11 Interface For VIS to show current Visitor 26
Figure 12 Manage Offices 27
Figure 13 Manage ID CARDS 27
Figure 14 Manage Regions 28
Figure 15 Manage Districts 28
Figure 16 System Users Activity 29
Figure 17 VIS Terms of Use Interface 29
Figure 18 Reporting User Interface Dashboard 30
Figure 19 Login code 32
Figure 20 Login Source Code 33
Figure 21 Login Source Code 33
Figure 22 Admin Dashboard Source Code 34
Figure 23 Admin Dashboard Source Code Continue 34
Figure 24 Admin Dashboard Source Code Continue 35
Figure 25 Users Dashboard Source Code 35
Figure 26 Users Dashboard Source Code Continue 36
Figure 27 Users Dashboard Source Code Continue 37
Figure 28 Users Dashboard Source Code Continue 38
Figure 29 Users Dashboard Source Code Continue 38

LIST OF ABBREVIATIONS

ABAC	-Attribute-based access control
CIA	-Confidentiality Integrity and Availability
VIS	-Visitor Information System
CSS	-Cascading Style Sheet
DAC	-Discretionary access control
ERD	-Entity Relationship Diagram
HTML	-Hyper Text Markup Language
HTTP	-Hypertext Transfer Protocol
IT	-Information Technology
MAC	-Mandatory access control

MYSQL	-My standard Query Language
PHP	-Hyper Text Preprocessor
RAM	-Random Access Memory
RBAC	-Role-based access control
RDBMS	-Relational Database Management Systems
SSO	-Single Sign-On
UCC	-University of Dar es salaam Computing Center
UCD	-user-centered design
UML	-Unified Modeling Language
WAMP	-Windows Apache MySQL PHP
XAMPP	-Cross platform Apache MySQL Perl and PHP

CHAPTER ONE

INTRODUCTION

1.1 Background of the project

A Visitor Information System (VIS) is an evolving concept defined as a systematic collection of visitors' information. It is a record in digital format that is theoretically capable of keeping track of a list of visitors and their details for effective service from both clients and the company. The project's goal is to improve the visitor's experience by providing timely and accurate information about the location, the event, and the surrounding areas. Additionally, the system can help venue managers and organizers to manage crowds, gather data on visitor behaviour, and enhance the overall visitor experience. The implementation of such systems has become increasingly popular in various public spaces, including museums, airports, and sports arenas.

This system will track the usage of a public office by gathering increasing amounts of information. A visitor information system can record the usage of the facilities by specific visitors and provide documentation of visitors' whereabouts. Since a visitor information system provides a record of the office used and the person visited, this system will often use to complement office security systems and access control systems

The case study for the Visitor Information System was conducted at the Tanzania Bureau of Services, which is a government agency responsible for regulating and providing services in various sectors, including trade and industry. The agency's mandate includes promoting Tanzania's products industry, which is a crucial contributor to the country's economy. The project aimed to develop an efficient and user-friendly information system to enhance the visitor's experience and provide accurate information on visitors. The system was designed to cater to both domestic and international visitors, capturing and storing all their information in a database, along with other relevant information. The project's success would improve efficiency for the visitor information system by streamlining the process of capturing and organizing visitor information accurately.

1.2 Background of the Organization

The Tanzania Bureau of Standards (TBS) is a government agency established in 1975 under the Ministry of Industry and Trade. Its primary mandate is to provide quality control and

standardization services in various sectors, including manufacturing, agriculture, and tourism. TBS's mission is to promote and facilitate the development of efficient and effective quality infrastructure systems that meet the needs of the public and industry. The organization aims to enhance the competitiveness of Tanzanian goods and services in the global market and safeguard the health and safety of the public. TBS has various departments, including standards development, certification and inspection, and metrology. The agency also collaborates with other government bodies, international organizations, and private sector stakeholders to achieve its objectives.

1.3 Statement of the problems

After careful analysis of the project the researcher observed that the following problems are mainly encountered by different organization. The present method of visitor information system is mainly based manual recording of visitor information, also the present system makes use of small or no database for recording of visitor information. Due to the lack of a comprehensive database, accountability is usually very difficult, the present system is time-consuming, and owing to the paper and pen method of recording information visitors may not give out the full information needed at a particular time queuing may occur at the cause of entering the information needed.

1.4.1 General Objectives

The general objective of this is to design and implement a web-driven database site to automate manual activity and keep a record of visitors' information for easy retrieval and annual report analysis.

1.4.2 Specific Objectives

1. To record and track visitor's information at Tanzania Bureau of Standards such as name of visitor, purpose of visit, check in time, checkout time.
2. To Design and develop system database to store Visitor information.
3. To Design and develop system interfaces to be user friendly
4. To Design and develop an easier way to filter record attributes

5. To provide instant report and status on every visit at Tanzania Bureau of Standards.

CHAPTER TWO

REQUIREMENTS ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish

2.1 Overview of the existing System

2.1.1 Activities of the system

Visitors must fill out paper forms with their personal information, purpose of visit, and other relevant details, which are then stored in filing cabinets or binders. This process is time-consuming, error-prone, and inefficient, and does not provide visitors with real-time information or a seamless experience.

2.1.2 Problem of existing system

Currently, the Tanzania Bureau of Standards (TBS) does not have a dedicated system in place for managing visitor information. All activities related to visitor management are performed manually, including registration, data collection so they keep entire information regarding the visitor in the visitor master file.

2.2 Over view of the proposed system

The proposed Visitor Information System will automate and streamline visitor management processes at TBS. The system will enable visitors to be registered, provide real-time information on TBS products, services, and facilities, and offer various communication channels for visitors to make inquiries or request assistance. The system will also track visitor behavior and preferences to optimize service delivery, marketing strategy, and overall visitor experience.

Generally, there has been a criterion to work on any job or task for a specific purpose. Nobody works without specific detailed information about the particular task he is performing. Thus, any

transaction can be performed either or check in and check out. In the computerized system, the first screen of the system would be a Dashboard and a list of menus.

These menus contain the options of collecting the information of any visitor and other System Administration settings. Any transaction can be performed in the visitor Information system i.e. routine processes. After the transaction is completed the user can log off from the system by simply quitting from the system. The main objective of the proposed system is to design and implement a web-driven database site to automate manual activity and keep a record of visitors' information for easy retrieval and annual report analysis. The system can be handy to the user in the following reasons

- To provide quick and efficient means for gathering the visitor information
- To automate the work such as gathering visitor information and allocating the zones they are allowed to visit.
- User-friendly interface for System user for visitor registration
- Secure login and authentication for employees
- Customizable reporting features for users

Features

The visitor Management system of the organization is developed to overcome the most of the problems occurring in the Manual System by computerizing the existing system. The features of the newly proposed computerized system are described in brief as below. After computerizing the system, the committee of the organization or the user of the system can finish their work in least amount of time and efforts. The computerized system has many gains and efforts, which the Manual system can't give, in any type of situations. In any Manual system if we take, the main problem arising is to maintain the number of records and finding a particular record. In Manual system the user has to note down that the check in and check out of each and every visitor but in the case of computerized system the user just has to enter once and automatically all the messages delivered by all the visitors will be displayed and thus the user does not have to worry because all the work done is automatically and the system itself. Computerized systems are most helpful in dealing with areas where database comes into the existence. A computer can hold large amount of data in its storage devices and it can operate at very high speed. The user can put the entire information in the computer and can be able to perform any type of task which when done Manual System is tedious and time consuming. The new system will have the facility to sort data according to any specific type on the basis of what the user wants in any order. Also, with the help of computerized system if the user wants to access any single user's data from many users' data he can automatically get the desired data of the desired visitor. In a fraction of second which is again time saving and very quick. Some of the features of the proposed system are given below

- Maintaining the database for the visitor, check-in and check-out, etc.
- Removal of data redundancy.
- Data consistency.
- Menu driven interface.

- Ensure data security.

2.2.1 Functional requirements

Functional requirements describe what the system should do and how it should behave. Some examples of functional requirements for the Visitor Information System include

- User Registration and Authentication: The system should allow users to register as visitors, either online or on-site, providing their personal information such as name, contact details, and purpose of visit.
- Visitor Tracking: The system should be able to track visitors throughout their stay, including their location, time-in and time-out times, and the activities they engage in while on-site.
- Reporting: The system should allow users to generate reports on visitor activities and patterns, providing insights into the popularity of specific areas, the length of stay, and any issues or concerns that visitors may have.
- Database Management: The system should maintain a database of visitor information, allowing for easy retrieval and updating of visitor records.
- Activity Tracking: The system should track user activity on the system, including logins, page views, and other interactions, to ensure the system is being used appropriately.
- Problem Identification: The system should track visitor reasons for visiting the company, allowing for identification of any issues or concerns that visitors may have and the potential for improvement.
- Security: The system should have appropriate security measures in place, including password protection, data encryption, and user access control, to prevent unauthorized access and ensure visitor privacy.
- Scalability: The system should be designed with scalability in mind, to accommodate future growth and changing business needs.

- User-Friendly Interface: The system should have a user-friendly interface, with clear navigation and easy-to-use features, to ensure maximum usability and satisfaction for visitors and employees.
- Customization: The system should be customizable to fit the specific needs of the business or organization, including the ability to add or remove features as necessary.

2.2.1.1 New Visitor Information System

I. The system will allow admin to add new system user into the system
II. The system will allow admin and staff to create new visitor
III. The system will allow admin and staff view and generate visitor report
IV. The System will allow admin to manage System Activity
V. The system Will allow admin to manage ID-CATEGORY

2.2.1.2 Visitor management

I. The system will enable staff user to add new visitor
II. The system will allow staff to update only his profile
III. The system will allow staff to generate visitor reports

2.2.1.3 Reporting management

I. system will allow Reporting User to view his or her profile only
II. system will allow Reporting User to generate Visitor Report
III. system will allow the Reporting User to update his or her profile

2.2.1.4 System administration

I. The system will allow an administrator to generate system reports
II. The system will allow an administrator to manage districts
III. The system will allow an administrator to manage id category
IV. The system will allow an administrator to manage office and its branch
V. The system will allow an administrator to manage user accounts
VI. The system will allow an administrator to manage regions

2.2.2 Non-functional requirements

Non-functional requirements explain and describe requirements that support the main functions of the system, but they are not part of the system functionalities. Some examples of non-functional requirements for the Visitor Information System include:

I. Fast response time for users of visitors Information System.

II. Scalability and flexibility to accommodate future updates and enhancements.

III. Backup and disaster recovery measures to prevent data loss.

IV. Usability and Accessibility for visitors and Employee.

V. Compliance with relevant standards and regulations.

2.2.2.1 Operational

Operational considerations are important for the Visitor Information System, as it will be used by TBS employees on a daily basis. Some examples of operational considerations include:

- Ease of use and navigation for employees
- Compatibility with existing systems and infrastructure
- Training and support for employees who will use the system
- User roles and permissions for data security and access control

2.2.2.2. Performance

Performance considerations are important for the Visitor Information System, as it will need to operate smoothly and efficiently. Some examples of performance considerations include:

- Fast response time for users of visitors Information System
- Scalability and flexibility to accommodate high volumes of traffic
- Minimal downtime for maintenance and updates

- Optimal use of system resources to minimize costs

2.2.2.3. Security

Performance considerations are important for the Visitor Information System, as it will need to operate smoothly and efficiently. Some examples of performance considerations include:

- Fast response time for users of visitors Information system
- Scalability and flexibility to accommodate high volumes of traffic
- Minimal downtime for maintenance and updates
- Optimal use of system resources to minimize costs

2.2.2.4 Cultural and Political

Cultural and political considerations are important for the Visitor Information System, as it will be used by visitors from different backgrounds and may be subject to local regulations. Some examples of cultural and political considerations include:

- Multi-lingual support for visitors who may not speak the official language
- Cultural sensitivity in user interface design and content
- Compliance with local data protection laws and regulations
- Accessibility for visitors with disabilities or special needs

2.2.3 System requirements (Hardware and software requirements)

2.2.3.1 SOFTWARES

I. XAMPP OR WAMPSERVER
II. Visual studio Code
III. Web Browser

2.2.3.2 HARDWARES

Computer with minimum 2GB RAM, Processor Intel insider @ 1.0GHZ speed or greater and storage capacity of at least 20GB with Window 7,8,10 or any Linux Operating System.

CHAPTER THREE

USE CASES

3.1 Use case Model

A use case (UML) is a description of how a person who actually uses that process or system will accomplish a goal. It's typically associated with software systems, but can be used in reference to any process. For example, imagine you're a cook who has a goal of preparing a grilled cheese sandwich. The use case would describe through a series of written steps how the cook would go about preparing that sandwich. A use case helps you understand where errors could occur in the process and design features to resolve those errors.

Three elements that a use case must contain:

1. Actor, which is the user, which can be a single person or a group of people, interacting with a process
2. System, which is the process that's required to reach the final outcome
3. Goal, which is the successful user outcome

3.1.1 Actors' specification

- **Admin:** The admin is the super user of the visitor information system and has full control over the system. The admin can perform the following activities:
  - Login
  - Manage users
  - Manage districts and regions
  - Manage roles and permissions
  - Manage identity cards used for the system
  - Manage office branches and offices
  - Perform database management
  - Manage logs for system accountability
  - Manage reports
  - Logout

- **Reporting User:** The reporting user is an actor of the visitor information system, who can generate reports and view visitor details. The reporting user can perform the following activities:
    - Login
    - Update profile
    - View visitor details
    - Generate reports
    - Manage Visit Purpose
    - Logout
- **Staff User:** The staff user is an actor of the visitor information system, who assists the client actor in ensuring that all the services are provided according to their roles. The staff user can perform the following activities:
    - Login
    - Manage visitors
    - Generate or manage reports
    - Update profile
    - Logout

3.1.2 Use Case Diagram

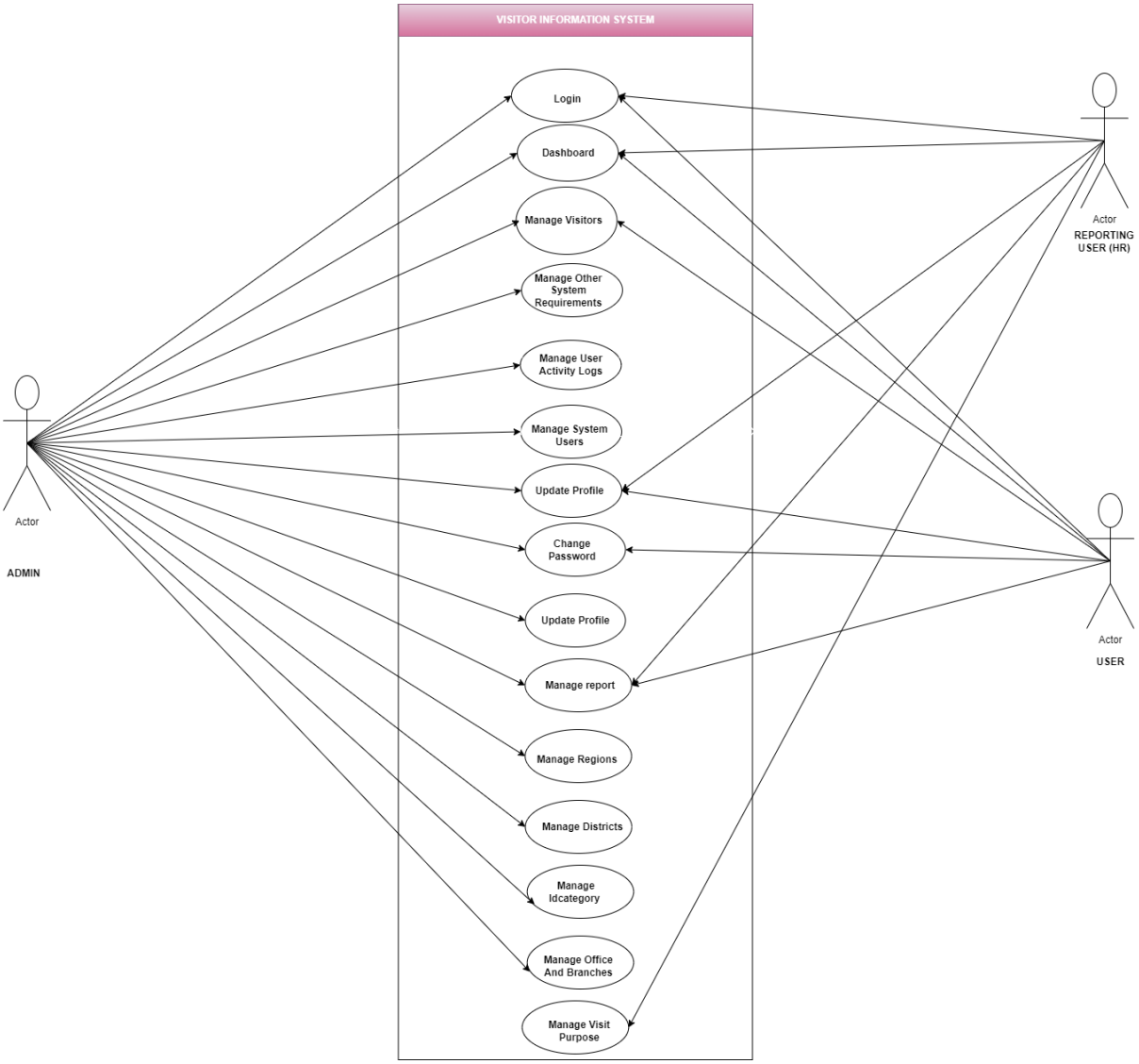


Figure 1 Use Case Diagram for Visitor Information System

3.1.3 Use Case Description

The use case description is a narrative document that describes, in general terms, the required functionality of the use case. Typically, it describes the use case goal and gives a general description of what usually happens, the normal course of events, adding a brief description of any minor variations.

The following tables will summary of each use case explanation

Actor Admin

Actor	Use case	Use case Descriptions
Admin	Manage visitor	This use case will allow the admin to add, edit, delete new visitor to the system
Admin	Manage users	This use case will allow the admin to Add, edit, and remove users and assign roles and deactivate. users(staff) from a blacklist to restrict access
Admin	Update profile	This use case will allow Admin to update his username, password, profile image only and his details
Admin	Manage Logs	This use case will allow Admin to Monitor user actions and ensure accountability
Admin	Manage Reports	This use case will allow Admin to Create, edit, delete, and view reports related to visitor information
Admin	Manage Locations	This use case will allow Admin to Manage Locations Add, edit, and delete locations where visitors can check in
Admin	Manage User Activity	This use case will allow Admin to track all system activity when user interact with the system.

Table 1 Admin use case.

Reporting User

Actor	Use case	Use case Descriptions
Reporting User	Manage report	This use case will allow the staff to generate and print data report for all visitor Information

Reporting User	Update Profile	This use case will allow user to update his username, password, profile image only and his details
Reporting User	Dashboard	This use case will allow reporting user to update his username, password, profile image only and his details
Reporting User	Manage Visit Purpose	This use case will allow reporting user to Manage all Visit Purpose

Table 2 Reporting User Use case

Actor User

Actor	Use case	Use case Descriptions
User	Manage Visitor	This use case will allow the user to view, edit, delete and to add a new visitor to the system
User	Update Profile	This use case will allow user to update his username, password, profile image only and his details
User	Generate Report	This use case will allow User to Create and view reports related to visitor information

Table 3 User use case.

CHAPTER FOUR

SYSTEM DESIGN

System design is an essential phase in the development of a visitor information system. The process involves defining the architecture, modules, and components of the system, as well as the interfaces and data that go through it. The ultimate goal of system design for the visitor information system is to meet the specific needs and requirements of the business or organization by engineering a coherent and efficient system. This includes designing the system to improve organizational processes and help employees perform their tasks more effectively. Additionally, the system design may also aim to satisfy stakeholders who are neither customers nor users of the system.

4.1 Design Goals

The goal of System Design is to design whole software, which fulfils all the requirements of customer. This leads to improve organizational systems, by applying software, which helps employees to perform business, tasks more effectively. A design may have goals to satisfy a stakeholder who is neither the customer nor the user.

4.2 System Decomposition

System decomposition is a critical aspect of system design. It involves organizing the system as a number of parts, which optimizes some metric related to program complexity, such as the modularity of the program or its maintainability. In the case of the visitor information system, the system decomposition can be achieved by dividing the system into various modules that interact with each other to provide visitors with a seamless experience.

Example of how system decomposition is done

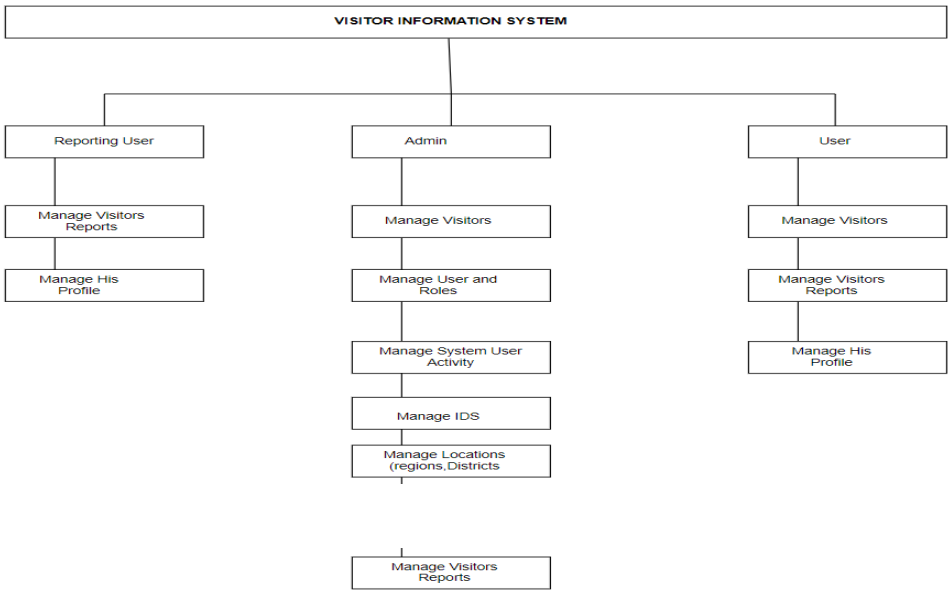


Figure 2 Functional ***Decomposition***

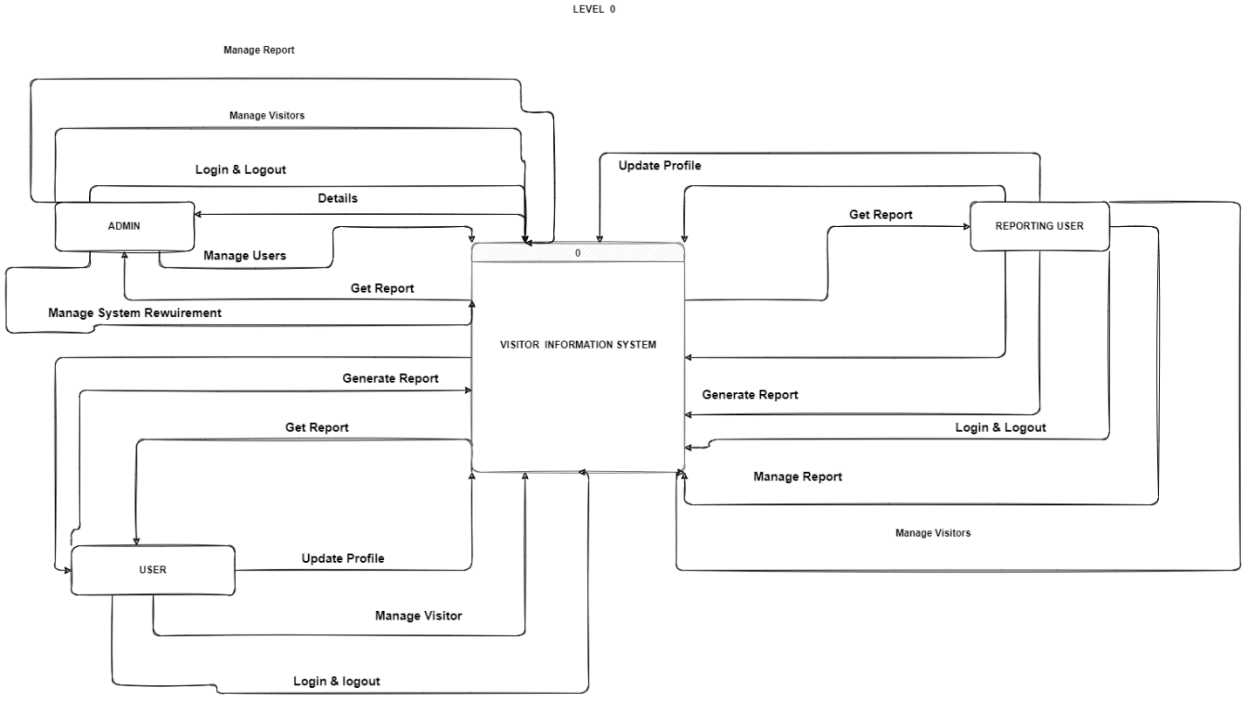


Figure 3 Context Diagram

4.3 Persistence data management

Persistence data management is another essential part of system design for the visitor information system. The persistent data stored in the system must be stable and recoverable. In this system, the database design must be demonstrated using an ERD, including normalization of the tables of the database from the first normal form to the third normal form. The normal form ensures that the database stores persistent data management for the intended system.

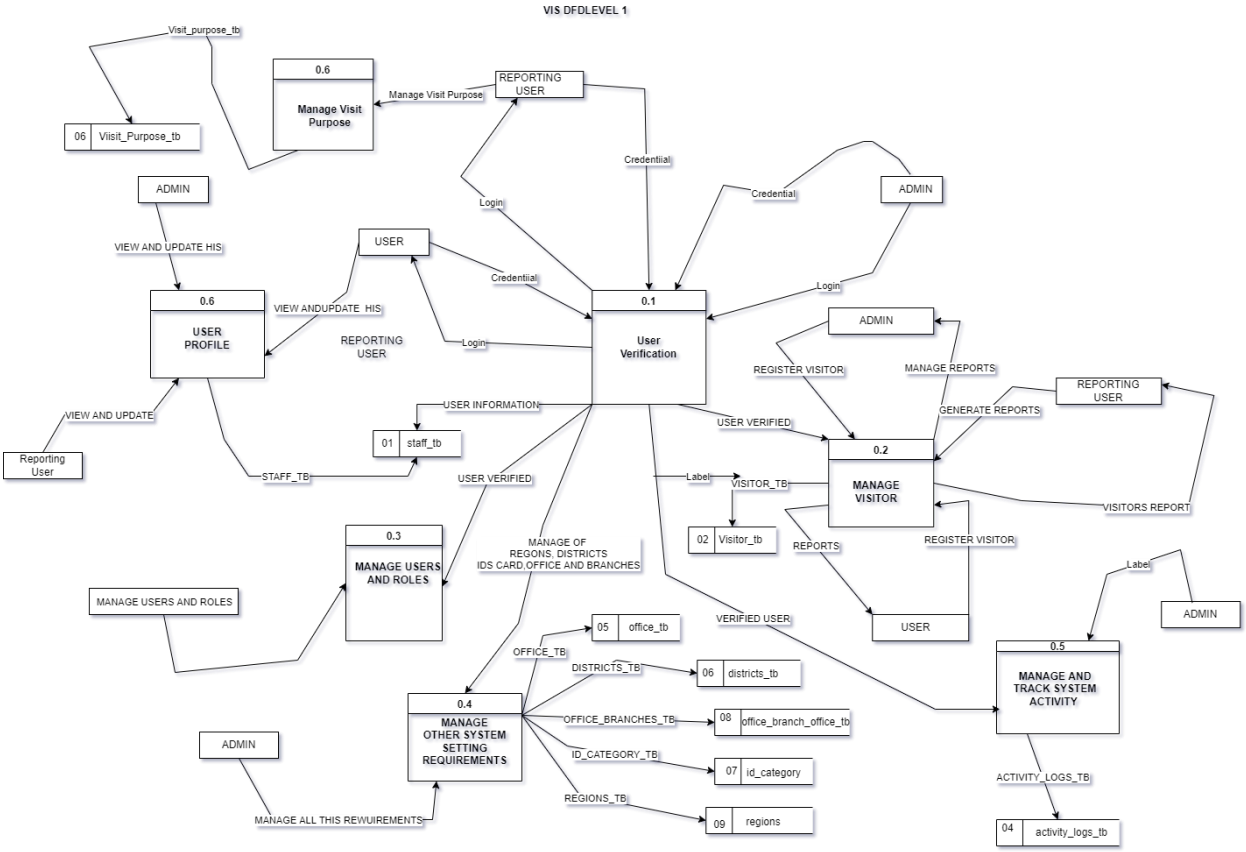


Figure 4 VIS DFD LEVEL 1

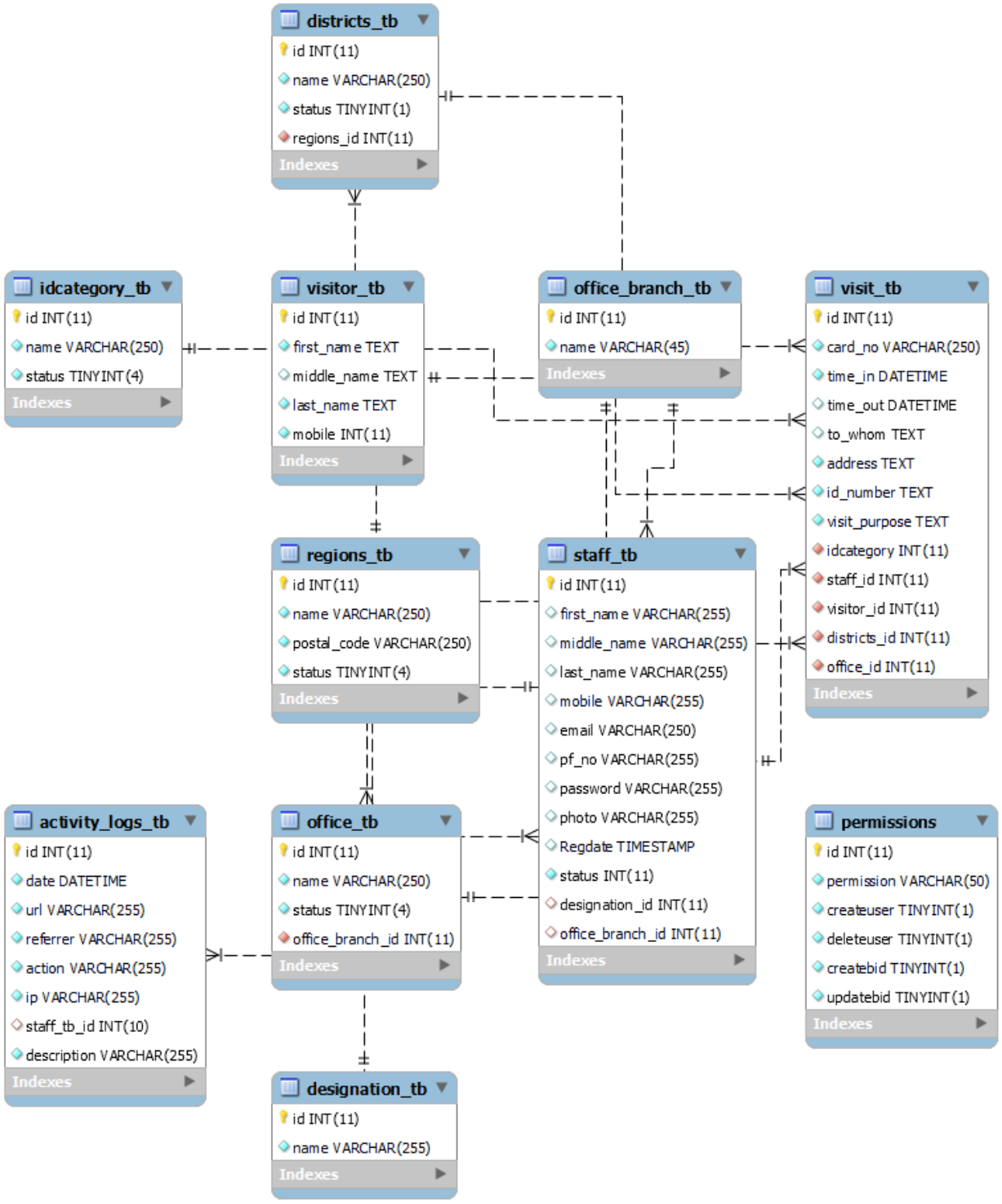


Figure 5 VIS Entity Relational Diagram

4.4 Access control and security

Access control minimizes the risk of authorized access to physical and computer systems, forming a foundational part of information security, data security and network security. Depending on your organization, access control may be a regulatory compliance requirement:

Access control and security are critical components of the system design for the visitor information system. Access control minimizes the risk of unauthorized access to physical and computer systems and forms a foundational part of information security, data security, and network security. For the visitor information system, access control can be implemented through the four main types of access control:

There are four main types access control. Organizations typically choose the method that makes the most sense based on their unique security and compliance requirements. The four access control models are:

- **Discretionary access control (DAC)**
  In this method, the owner or administrator of the protected system, data, or resource sets the policies for who is allowed access.
- **Mandatory access control (MAC)**
  In this nondiscretionary model, people are granted access based on an information clearance. A central authority regulates access rights based on different security levels. It's common in government and military environments.
- **Role-based access control (RBAC)**
  RBAC grants access based on defined business functions rather than the individual user's identity. The goal is to provide users with access only to data that's been deemed necessary for their role within the organizations. This widely used method is based on a complex combination of role assignments, authorizations, and permissions.
- **Attribute-based access control (ABAC)**
  In this dynamic method, access is based on a set of attributes and environmental conditions, such as time of day and location, assigned to both users and resources.
- **Why is access control important?**

The implementation of access control ensures that confidential information, including customer data, personally identifiable information, and intellectual property, from falling into the wrong hands. Without a robust access control policy, organizations risk data leakage from both internal and external sources.

It's particularly important for organizations with hybrid, multi-cloud cloud environments, where resources, apps, and data reside both on premises and in the cloud. Access control can provide these environments with more robust access security beyond single sign-on (SSO).

4.5 User Interface design

Finally, the user interface design is an essential part of system design for the visitor information system. The aim of user interface design is to design user interfaces for the system that increase usability and improve the user experience. The user interface design for the visitor information system must be easy to use and navigate, ensuring visitors can quickly find the information they need. A successful user interface design should achieve user-centered design (UCD) and ensure the visitors can accomplish their goals with ease.

The following are the main interface of the Visitor information system (VIS).

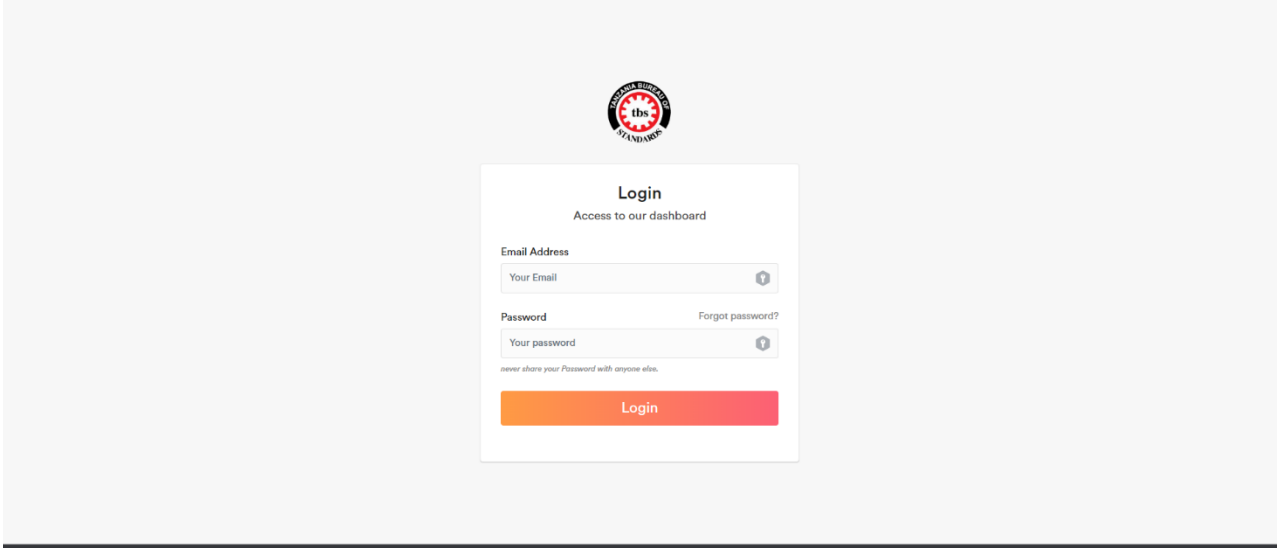


Figure 6 login form for vis

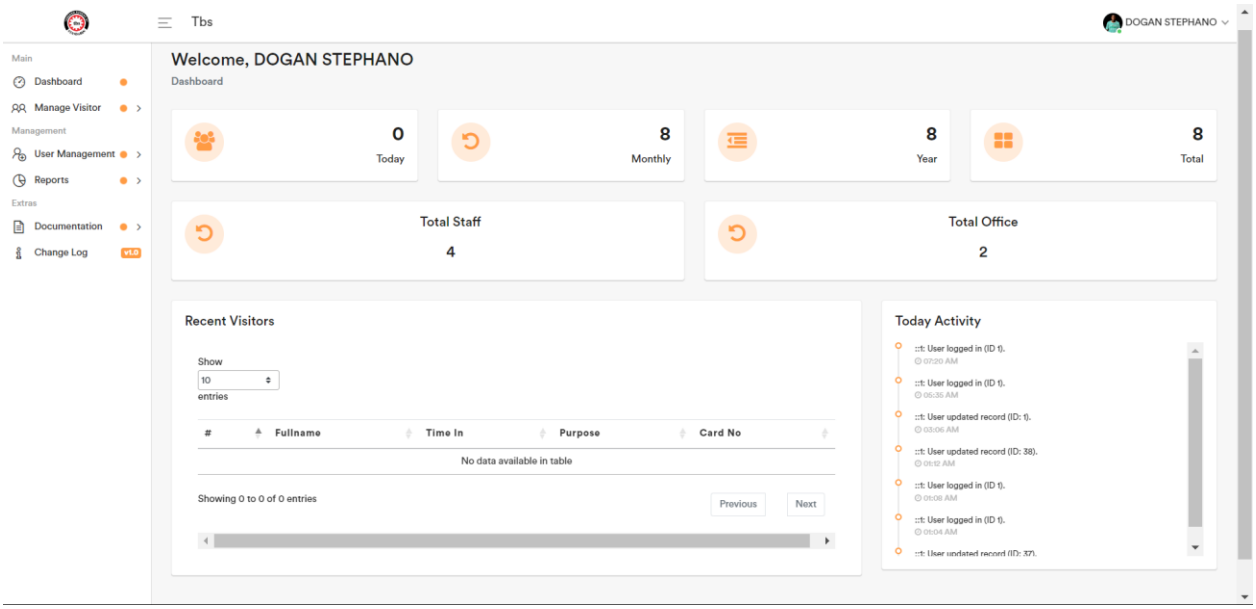


Figure 7 VIS admin Dashboard

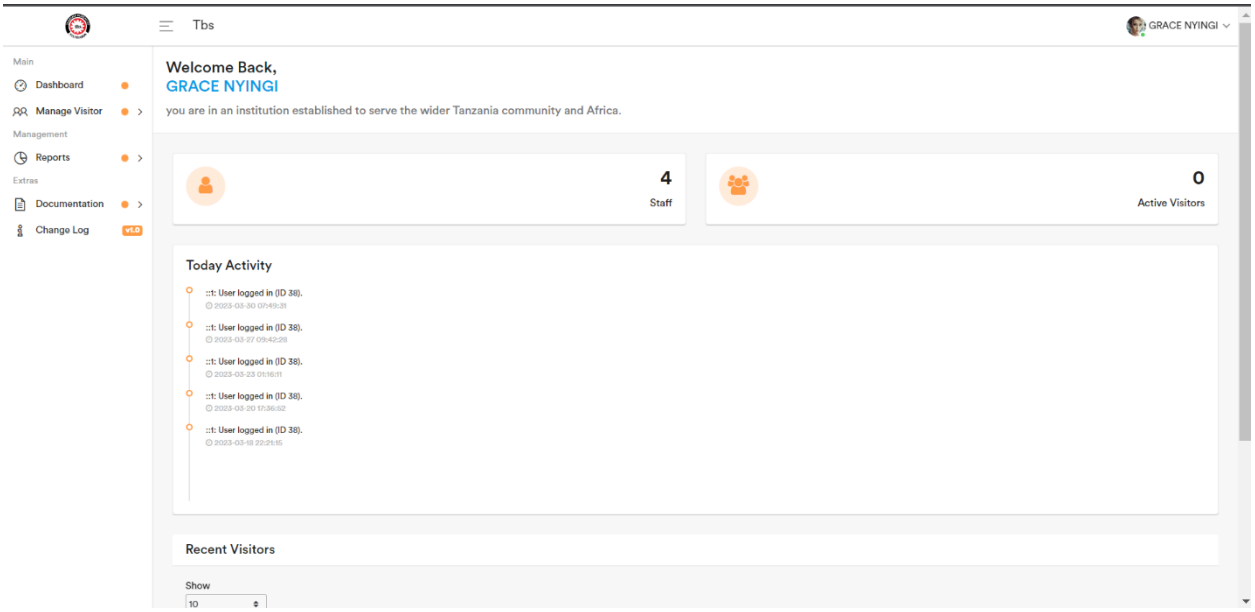


Figure 8 VIS User Interface

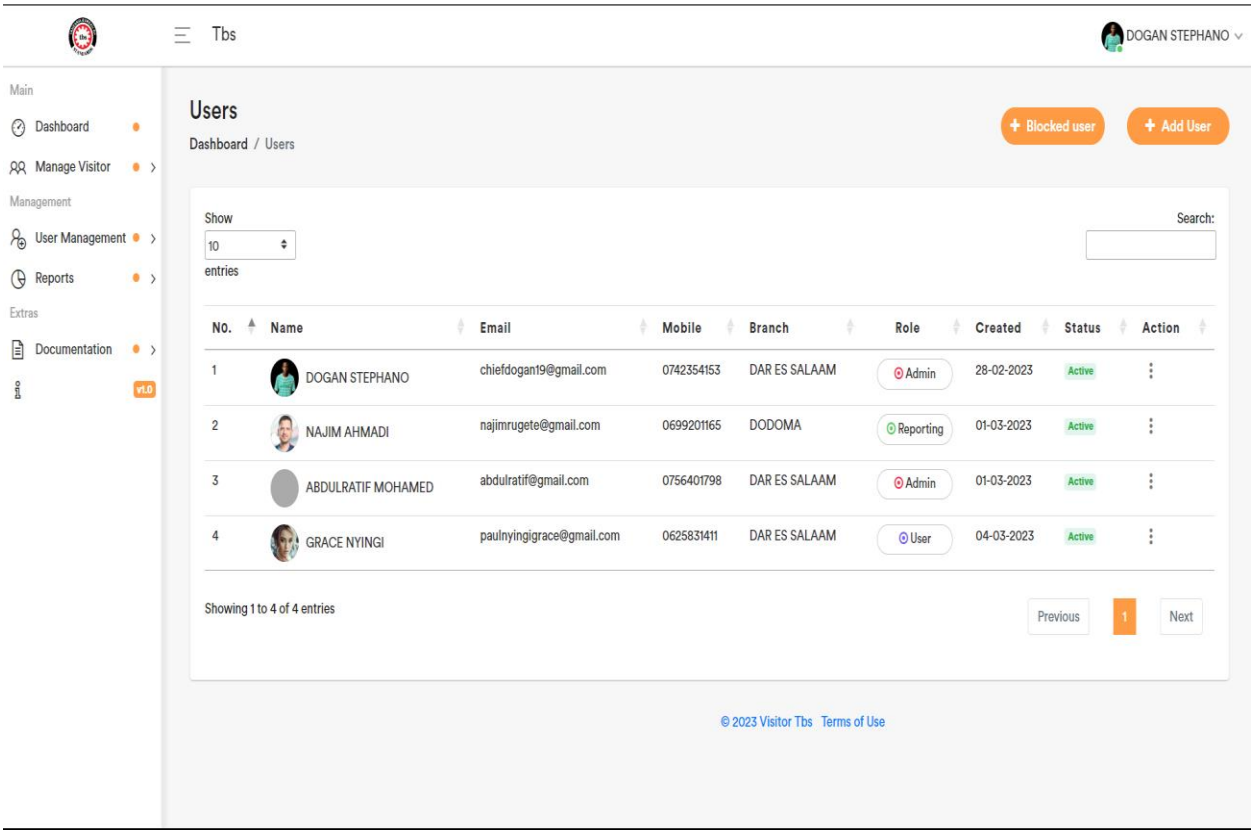


Figure 9 Vis Users Management View

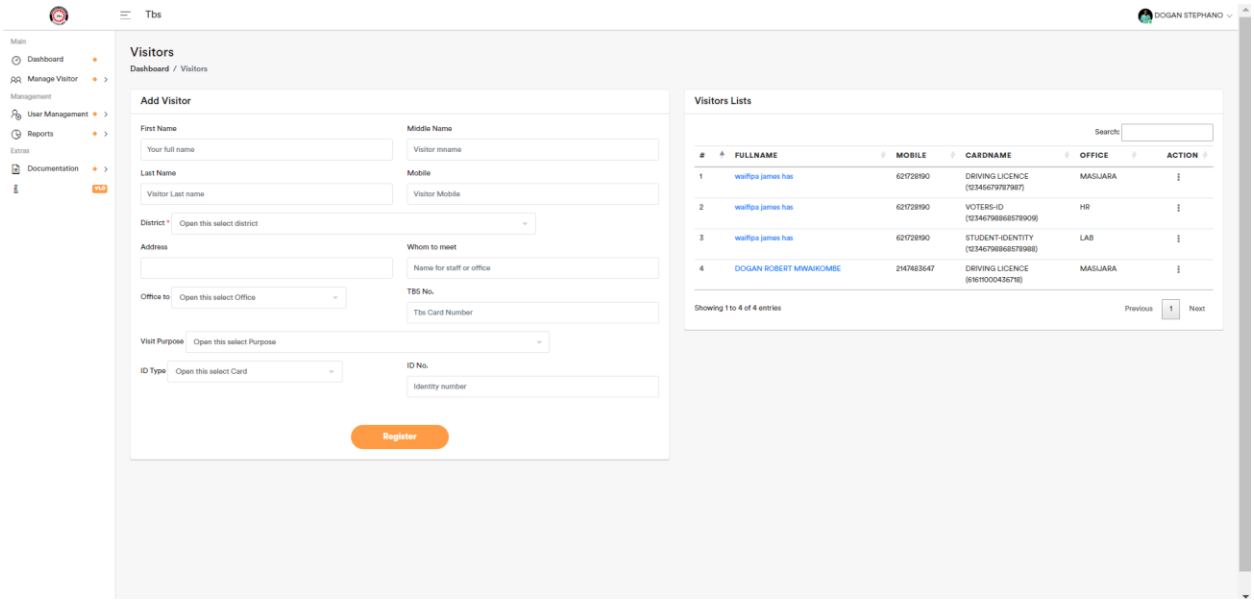


Figure 10 Interface For Visitor Registration

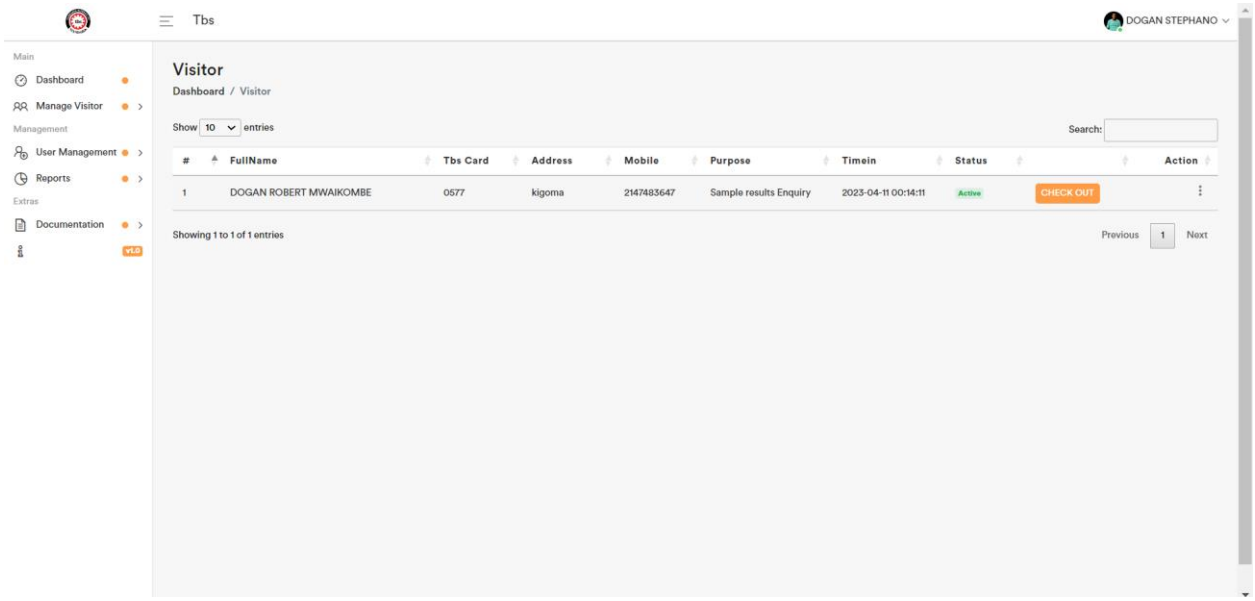
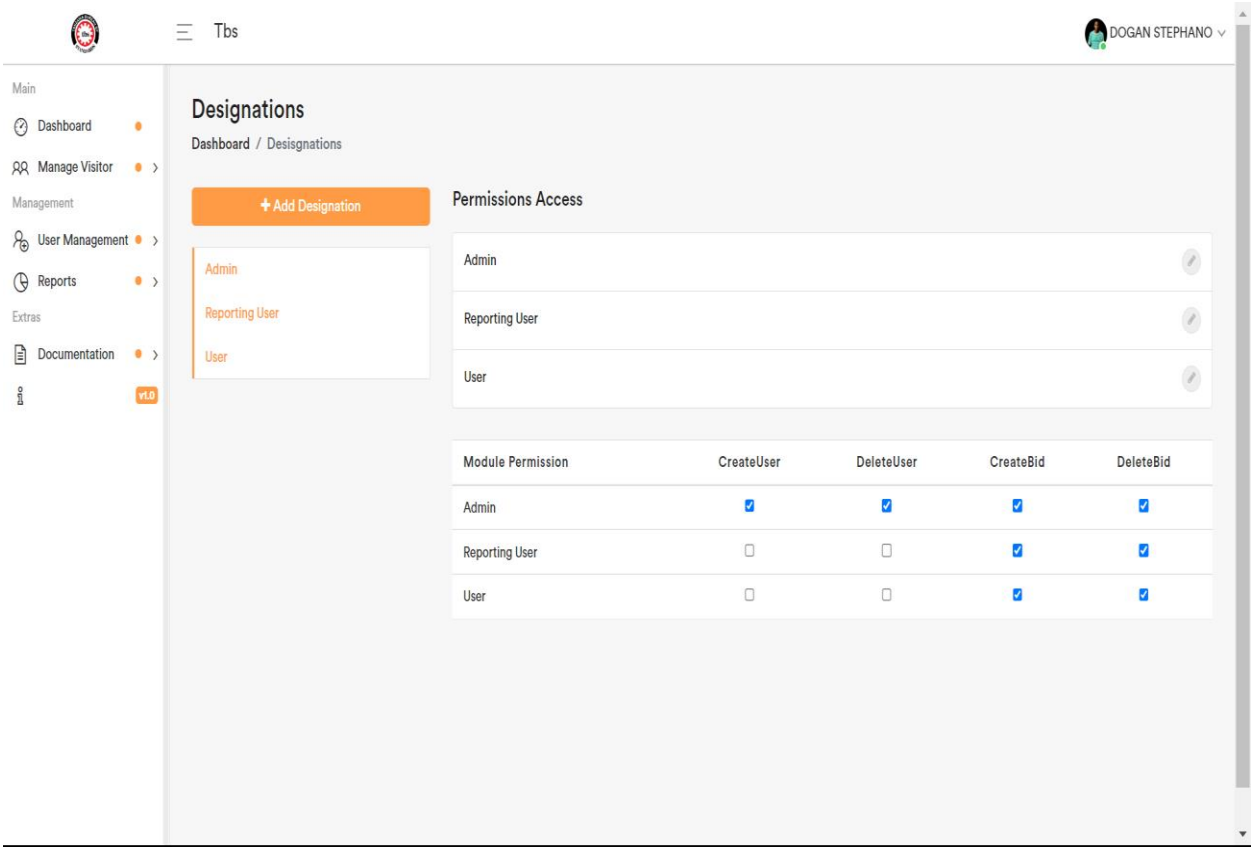


Figure 11 Interface For VIS to show current Visitor



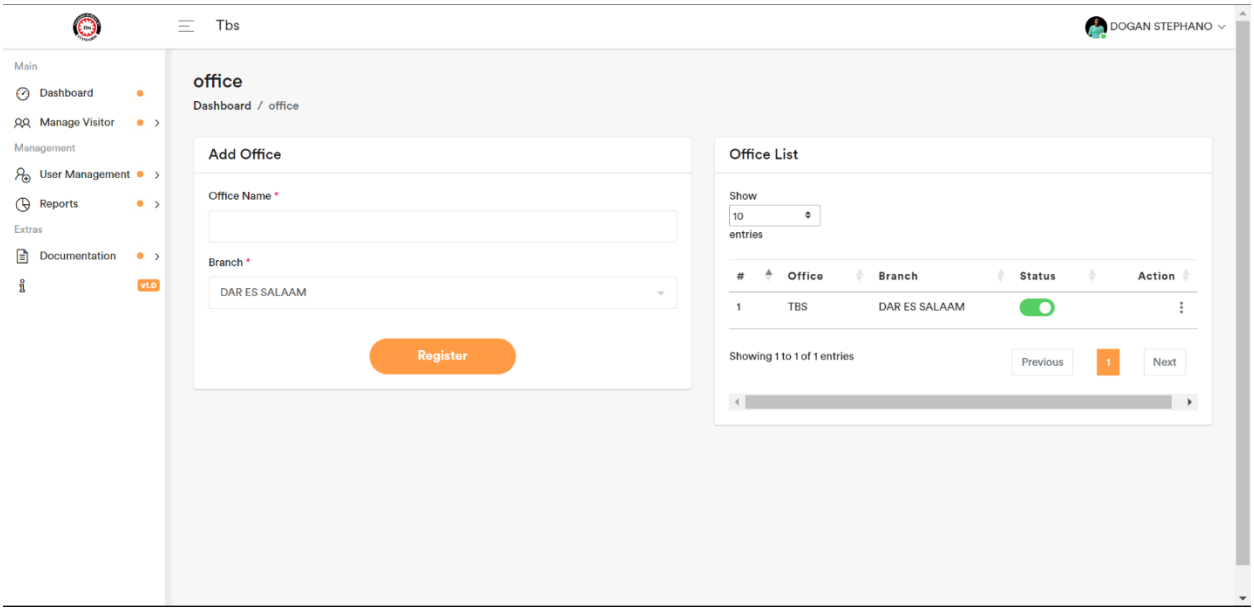


Figure 12 Manage Offices

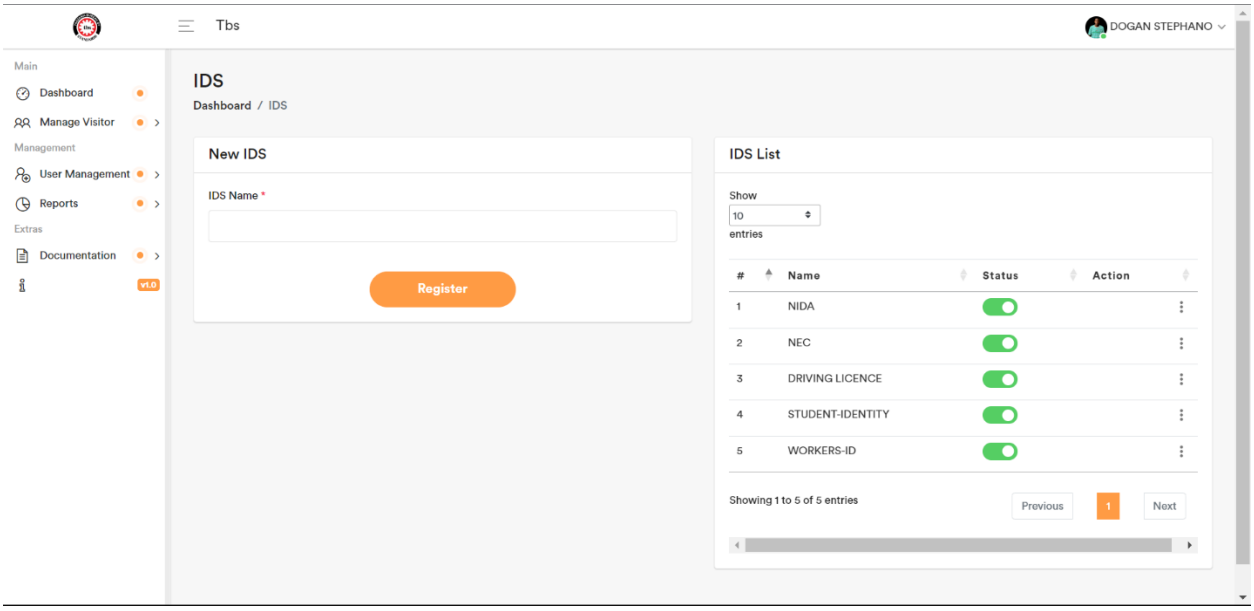


Figure 13 Manage ID CARDS

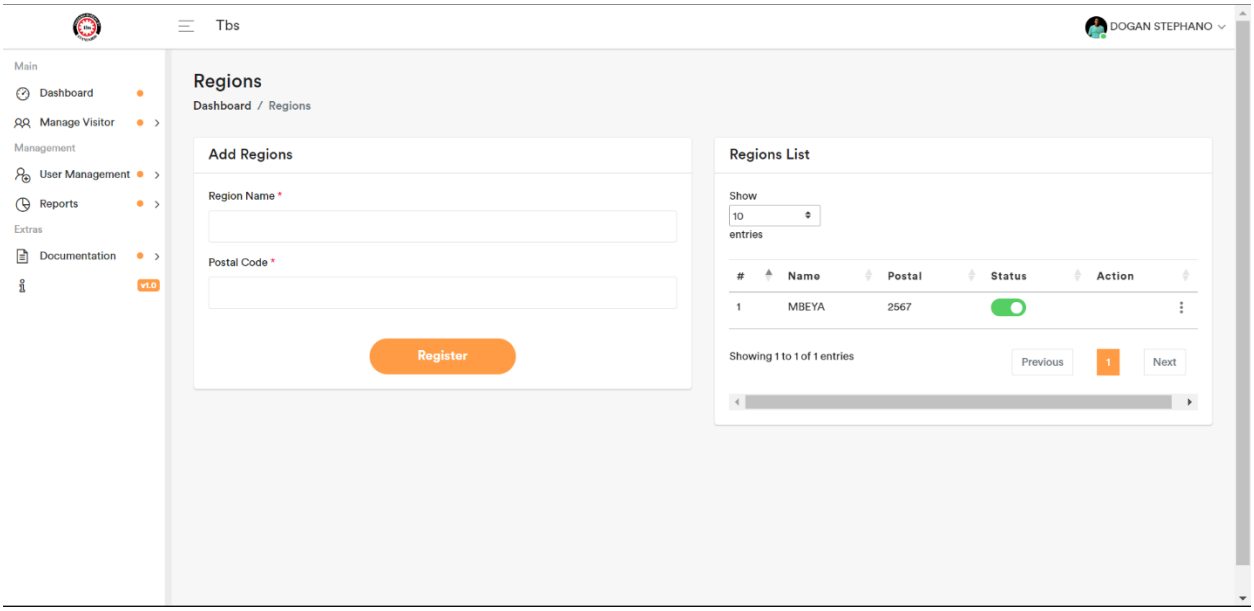


Figure 14 Manage Regions

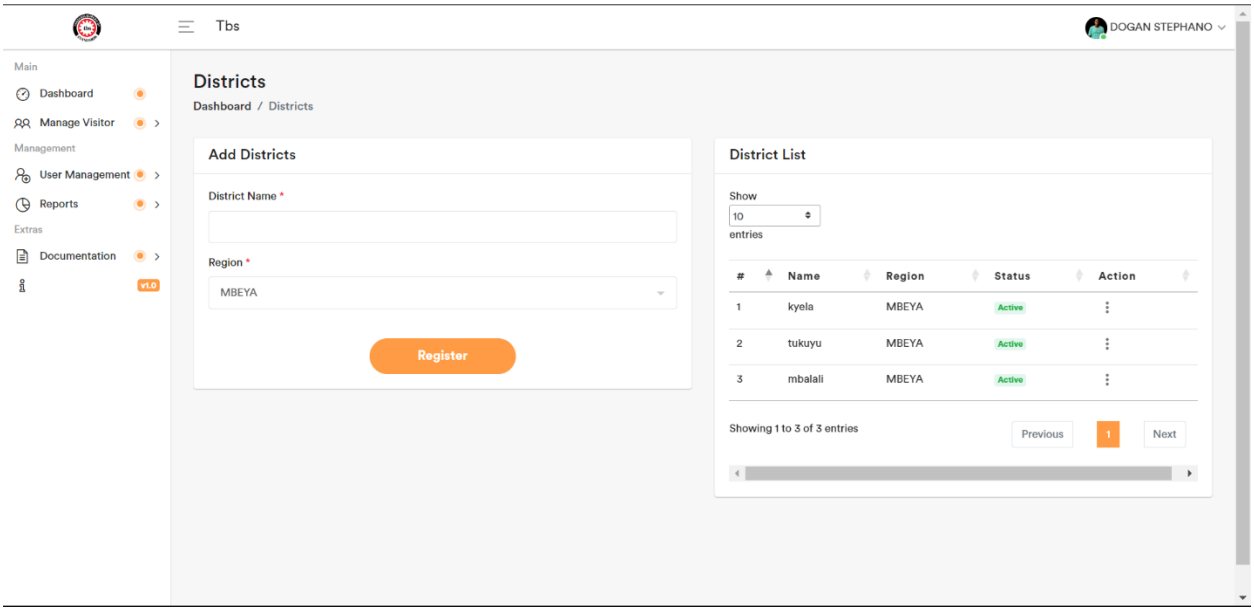
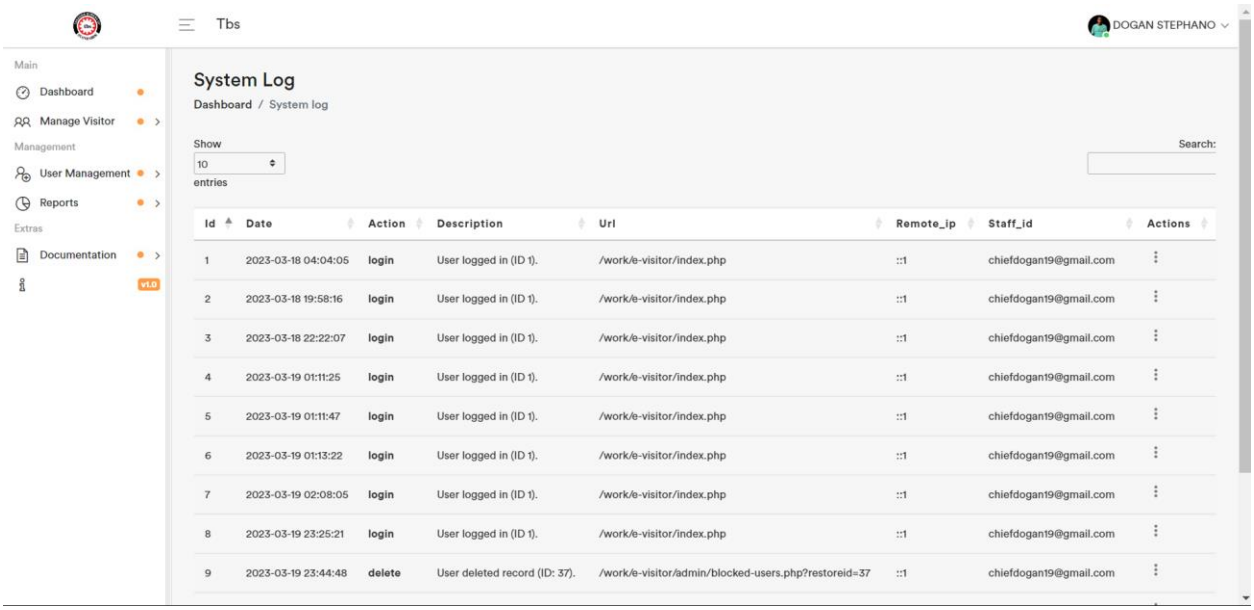


Figure 15 Manage Districts



Id	Date	Action	Description	Url	Remote_ip	Staff_id	Actions
1	2023-03-18 04:04:05	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
2	2023-03-18 19:58:16	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
3	2023-03-18 22:22:07	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
4	2023-03-19 01:11:25	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
5	2023-03-19 01:11:47	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
6	2023-03-19 01:13:22	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
7	2023-03-19 02:08:05	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
8	2023-03-19 23:25:21	login	User logged in (ID 1).	/work/e-visitor/index.php	::1	chiefdogan19@gmail.com	
9	2023-03-19 23:44:48	delete	User deleted record (ID: 37).	/work/e-visitor/admin/blocked-users.php?restoreid=37	::1	chiefdogan19@gmail.com	

Figure 16 System Users Activity

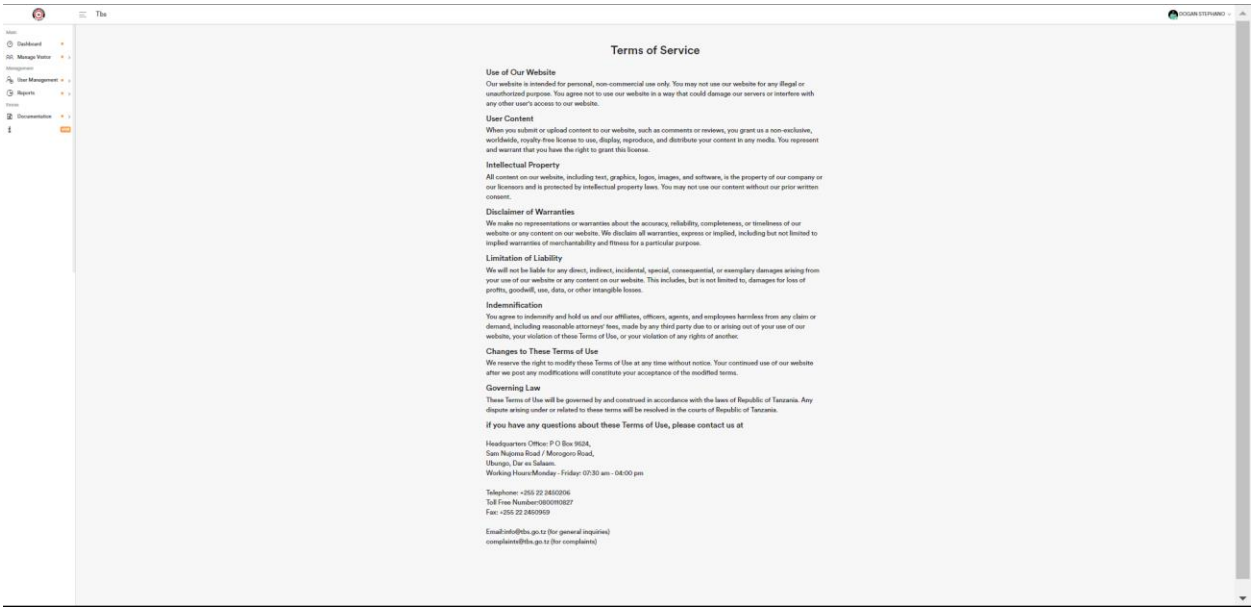


Figure 17 VIS Terms of Use Interface

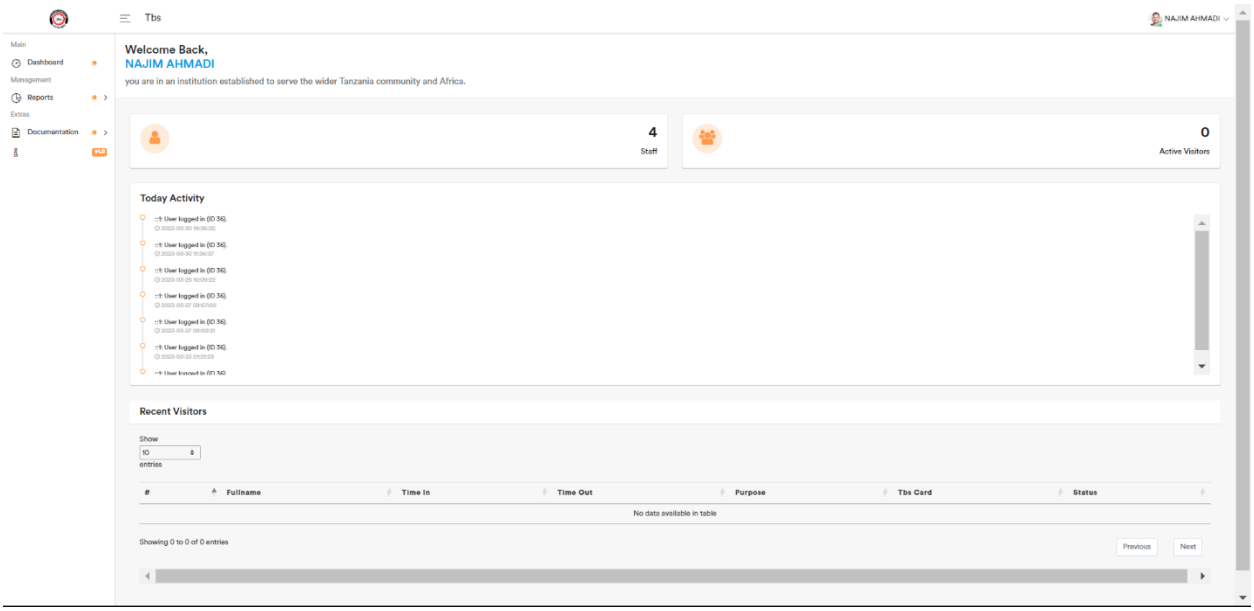


Figure 18 Reporting User Interface Dashboard

CHAPTER FIVE

CONCLUSION, EVALUATION AND FURTHER WORK

5.1 Conclusion

The increased number of visitors to various tourist sites has led to the need for a more efficient system to manage visitor information. The manual system for managing visitor information is prone to errors such as duplicating data, inaccurate data recording, and other consequences that may arise as a result. Implementing an automated system like the Visitor Information System (VIS) will help to overcome these challenges and provide a more user-friendly and GUI-oriented system for managing visitor information.

Despite the fact that implementing and maintaining automated systems may incur additional costs, the benefits of the VIS outweigh the cost of implementing and maintaining it. The VIS is designed to cater to the needs of people who manage various tourist sites and activities. It helps to manage visitor information more efficiently and effectively.

5.2 What we have achieved

Through the processes and procedures undertaken to develop the VIS, we have been able to overcome barriers and challenges that we encountered. We have gained more knowledge through

system designing, system analysis, and web-driven processes, which have helped us meet the project's objectives.

We have developed a functional and user-friendly VIS that will provide an efficient system for managing visitor information. The VIS is designed to be scalable and can accommodate the needs of various tourist sites.

5.3 Further Work

we recommend the implementation of a user IDS card verification that are valid for visitors to gain access to the premises or organization. This will enhance security and reduce the risk of unauthorized entry. Additionally, we would like to recommend the implementation of the visitor feedback rating feature that will allow visitors to provide valuable feedback to improve the quality of services.

References

Adirondack Regional Tourism Council, “Adirondacks USA Tourism Marketing Report 2021.” https://www.visitadirondacks.com/files/2021-adirondacks-usa-tourism-marketing-plan-final.pdf

G. Chauhan and K. Goel, "Visitor Management System for Tourist Places," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Allahabad, India, 2018, pp. 17-20. doi: 10.1109/ICACCCN.2018.8741522

Liu, Y.M., Y.D. Dong, and J. Wu. Scenic Management System Based on Number of Visitors. in Advanced Materials Research. 2014. Trans Tech Publ.

https://signinenterprise.com/wp-content/uploads/VMS-report-2020.pdf

Rodrigues, R., et al., Smart Gate Pass Security Management System Using Random Key Generation. International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN, 2021.

https://www.naturalspublishing.com/files/published/6p3jg121583s5s.pdf

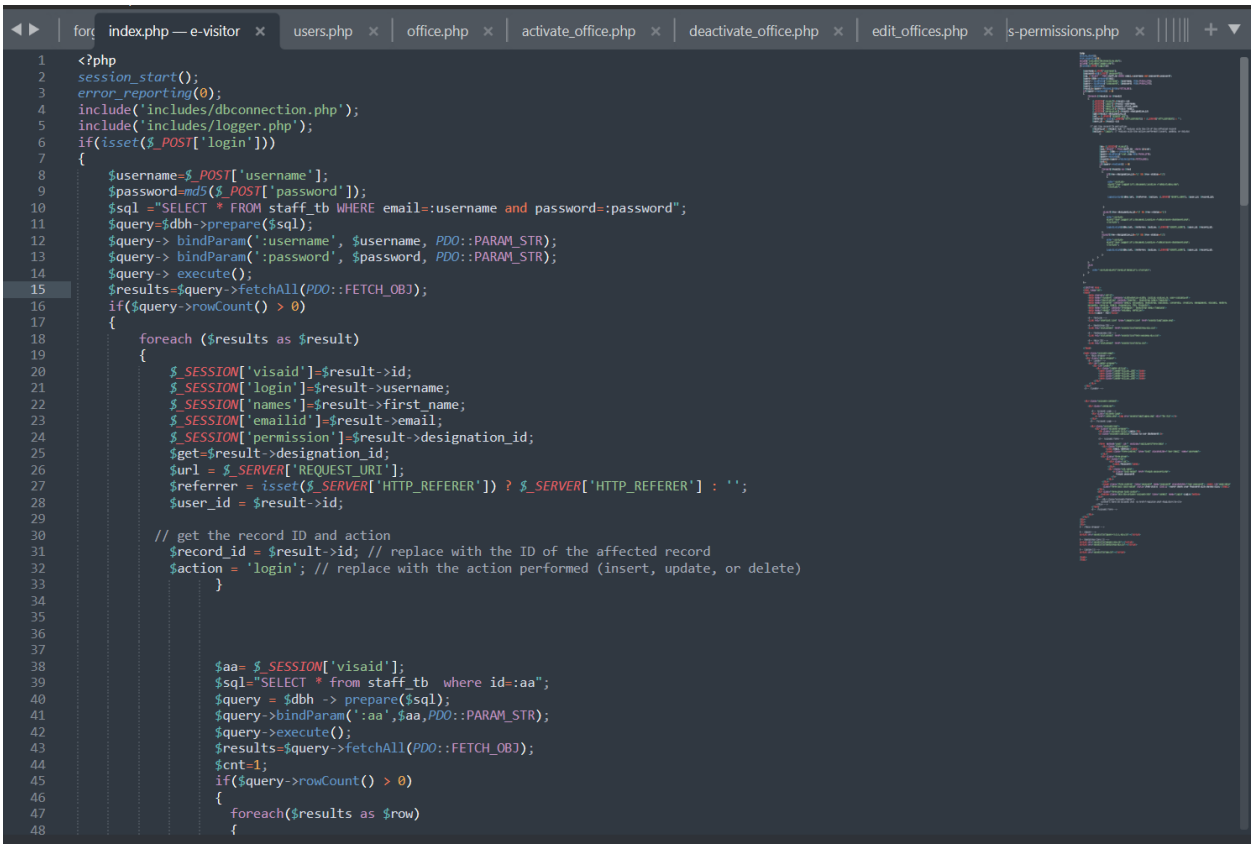
Appendices

Appendices 1:1 Login Source Code

```
                $aa= $_SESSION['visaid'];
                $sql="SELECT * from staff_tb  where id=:aa";
                $query = $dbh -> prepare($sql);
                $query->bindParam(':aa',$aa,PDO::PARAM_STR);
                $query->execute();
                $results=$query->fetchAll(PDO::FETCH_OBJ);
                $cnt=1;
                if($query->rowCount() > 0)
                {
                  foreach($results as $row)
                  {
                      if($row->designation_id=="1" && $row->status=="1")
                      {

                       echo "<script>
                       alert('Your logged in');document.location ='admin/index.php';
                       </script>";



                       logActivity6($dbh,$url, $referrer, $action, $_SERVER['REMOTE_ADDR'], $user_id, $record_id);



                    }

                    elseif($row->designation_id=="2" && $row->status=="1")
                    {
                       echo "<script>
                       alert('Your logged in');document.location ='admin/users-dashboard.php';
                       </script>";

                       logActivity6($dbh,$url, $referrer, $action, $_SERVER['REMOTE_ADDR'], $user_id, $record_id);

                    }
                    elseif($row->designation_id=="3" && $row->status=="1")
                    {
                       echo "<script>
                       alert('Your logged in');document.location ='admin/users-dashboard.php';
                       </script>";

                       logActivity6($dbh,$url, $referrer, $action, $_SERVER['REMOTE_ADDR'], $user_id, $record_id);

                    }
                }
            }
        }
```

Spaces: 4 PHP

Figure 19 Login code



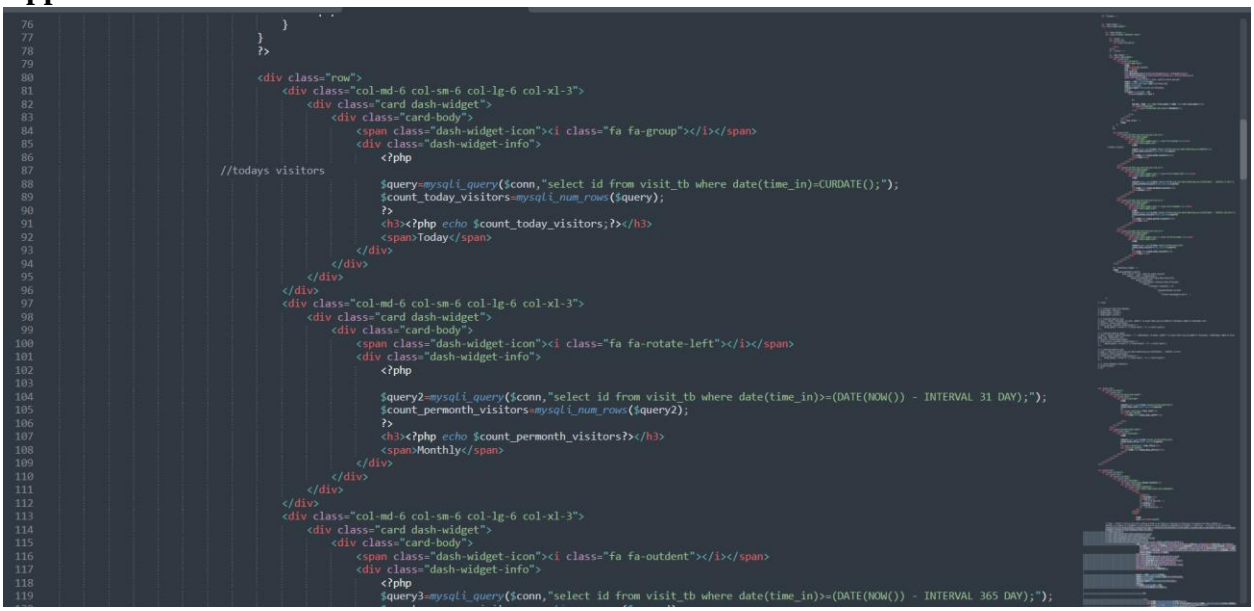

```php
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
include('includes/logger.php');
if(isset($_POST['login']))
{
    $username=$_POST['username'];
    $password=md5($_POST['password']);
    $sql ="SELECT * FROM staff_tb WHERE email=:username and password=:password";
    $query=$dbh->prepare($sql);
    $query-> bindParam(':username', $username, PDO::PARAM_STR);
    $query-> bindParam(':password', $password, PDO::PARAM_STR);
    $query-> execute();
    $results=$query->fetchAll(PDO::FETCH_OBJ);
    if($query->rowCount() > 0)
    {
        foreach ($results as $result)
        {
            $_SESSION['visaid']=$result->id;
            $_SESSION['login']=$result->username;
            $_SESSION['names']=$result->first_name;
            $_SESSION['emailid']=$result->email;
            $_SESSION['permission']=$result->designation_id;
            $get=$result->designation_id;
            $url = $_SERVER['REQUEST_URI'];
            $referrer = isset($_SERVER['HTTP_REFERER']) ? $_SERVER['HTTP_REFERER'] : '';
            $user_id = $result->id;

          // get the record ID and action
            $record_id = $result->id; // replace with the ID of the affected record
            $action = 'login'; // replace with the action performed (insert, update, or delete)
                  }




                  $aa= $_SESSION['visaid'];
                  $sql="SELECT * from staff_tb  where id=:aa";
                  $query = $dbh -> prepare($sql);
                  $query->bindParam(':aa',$aa,PDO::PARAM_STR);
                  $query->execute();
                  $results=$query->fetchAll(PDO::FETCH_OBJ);
                  $cnt=1;
                  if($query->rowCount() > 0)
                  {
                    foreach($results as $row)
                    {
```

Figure 20 Login Source Code

```php
<div class="row">
    <div class="col-md-12">
        <div class="row">
            <div class="col-md-6 text-center">
                <div class="card">
                    <div class="card-body">
                        <?php

                        $query2=mysqli_query($conn,"select id from staff_tb;");
                        $count_total_staff=mysqli_num_rows($query2);
                        ?>
                        <h3 class="card-title">Total Staff</h3>
                        <div id="bar-chartds">
                            <h3><?php echo $count_total_staff?></h3>


                        </div>
                    </div>
                </div>
            </div>
            <div class="col-md-6 text-center">
                <div class="card">
                    <div class="card-body">
                        <?php

                        $query2=mysqli_query($conn,"select id from office_tb");
                        $count_total_office=mysqli_num_rows($query2);
                        ?>
                        <h3 class="card-title"> Total Office</h3>
                        <div id="line-chartsd">
                            <h3><?php echo $count_total_office;?></h3>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>


<div class="row">
```

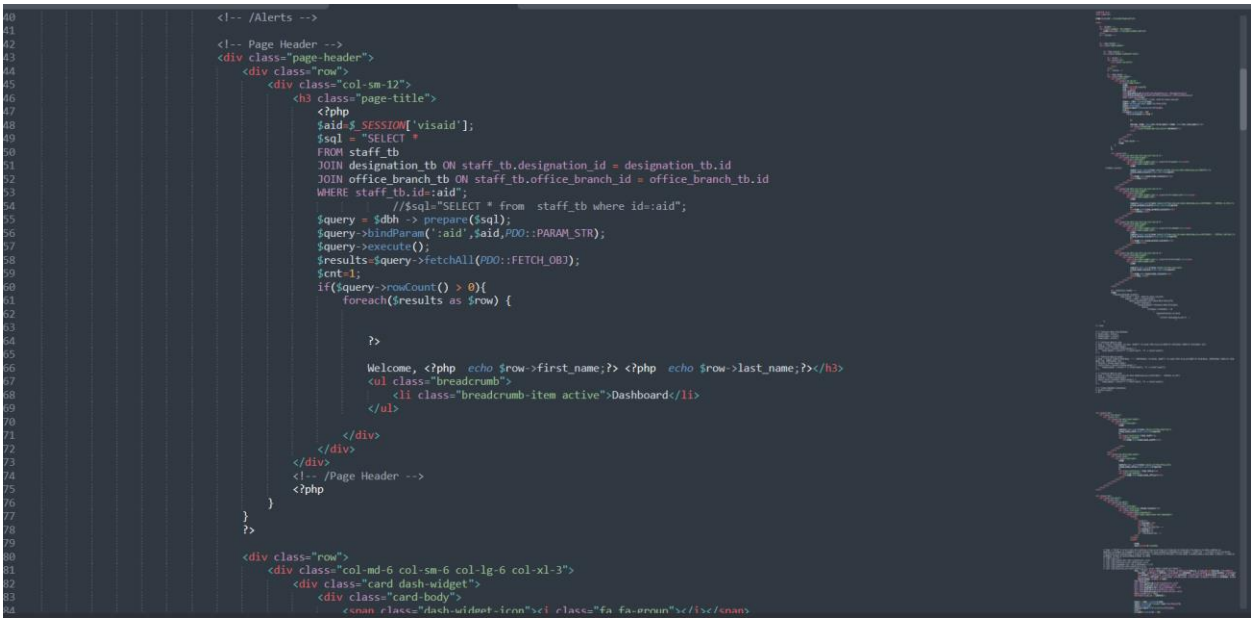
Figure 21 Login Source Code

Appendices 1:2 Admin Dashboard Source Codes



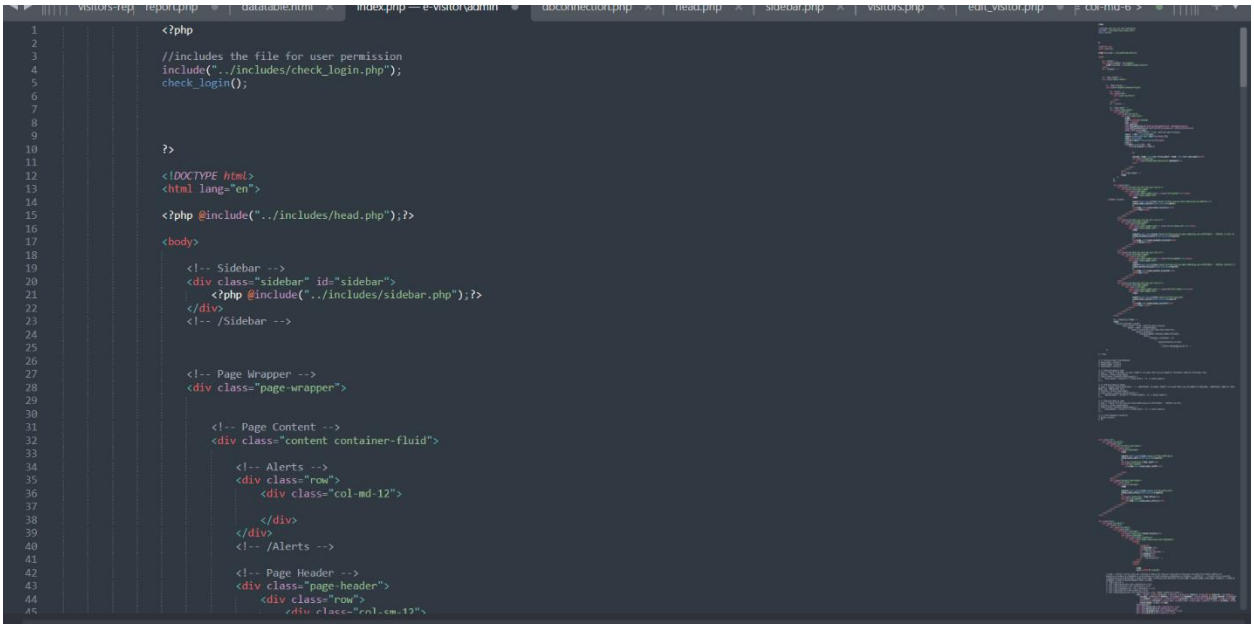
```
                    }
                }
                ?>

                <div class="row">
                    <div class="col-md-6 col-sm-6 col-lg-6 col-xl-3">
                        <div class="card dash-widget">
                            <div class="card-body">
                                <span class="dash-widget-icon"><i class="fa fa-group"></i></span>
                                <div class="dash-widget-info">
                                    <?php
//todays visitors
                                    $query=mysqli_query($conn,"select id from visit_tb where date(time_in)=CURDATE();");
                                    $count_today_visitors=mysqli_num_rows($query);
                                    ?>
                                    <h3><?php echo $count_today_visitors;?></h3>
                                    <span>Today</span>
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-6 col-sm-6 col-lg-6 col-xl-3">
                        <div class="card dash-widget">
                            <div class="card-body">
                                <span class="dash-widget-icon"><i class="fa fa-rotate-left"></i></span>
                                <div class="dash-widget-info">
                                    <?php

                                    $query2=mysqli_query($conn,"select id from visit_tb where date(time_in)>=(DATE(NOW()) - INTERVAL 31 DAY);");
                                    $count_permonth_visitors=mysqli_num_rows($query2);
                                    ?>
                                    <h3><?php echo $count_permonth_visitors?></h3>
                                    <span>Monthly</span>
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-6 col-sm-6 col-lg-6 col-xl-3">
                        <div class="card dash-widget">
                            <div class="card-body">
                                <span class="dash-widget-icon"><i class="fa fa-outdent"></i></span>
                                <div class="dash-widget-info">
                                    <?php
                                    $query3=mysqli_query($conn,"select id from visit_tb where date(time_in)>=(DATE(NOW()) - INTERVAL 365 DAY);");
```

Figure 22 Admin Dashboard Source Code



```
<!-- /Alerts -->

<!-- Page Header -->
<div class="page-header">
    <div class="row">
        <div class="col-sm-12">
            <h3 class="page-title">
                <?php
                $aid=$_SESSION['visaid'];
                $sql = "SELECT *
                FROM staff_tb
                JOIN designation_tb ON staff_tb.designation_id = designation_tb.id
                JOIN office_branch_tb ON staff_tb.office_branch_id = office_branch_tb.id
                WHERE staff_tb.id=:aid";
                            //$sql="SELECT * from  staff_tb where id=:aid";
                $query = $dbh -> prepare($sql);
                $query->bindParam(':aid',$aid,PDO::PARAM_STR);
                $query->execute();
                $results=$query->fetchAll(PDO::FETCH_OBJ);
                $cnt=1;
                if($query->rowCount() > 0){
                    foreach($results as $row) {


                        ?>

                        Welcome, <?php  echo $row->first_name;?> <?php  echo $row->last_name;?></h3>
                        <ul class="breadcrumb">
                            <li class="breadcrumb-item active">Dashboard</li>
                        </ul>

                    </div>
                </div>
            </div>
            <!-- /Page Header -->
            <?php
        }
    }
    ?>

    <div class="row">
        <div class="col-md-6 col-sm-6 col-lg-6 col-xl-3">
            <div class="card dash-widget">
                <div class="card-body">
                    <span class="dash-widget-icon"><i class="fa fa-group"></i></span>
```

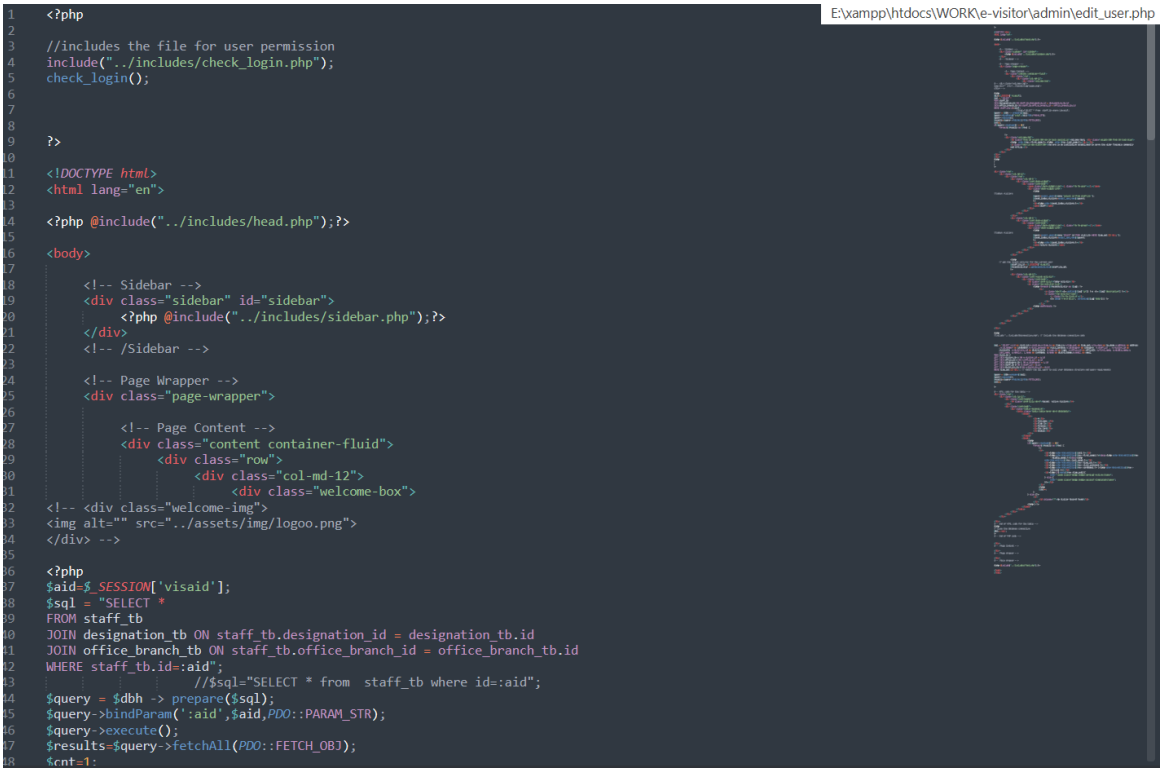
Figure 23 Admin Dashboard Source Code Continue



```php
<?php

//includes the file for user permission
include("../includes/check_login.php");
check_login();



?>

<!DOCTYPE html>
<html lang="en">

<?php @include("../includes/head.php");?>

<body>

    <!-- Sidebar -->
    <div class="sidebar" id="sidebar">
        <?php @include("../includes/sidebar.php");?>
    </div>
    <!-- /Sidebar -->



    <!-- Page Wrapper -->
    <div class="page-wrapper">


        <!-- Page Content -->
        <div class="content container-fluid">

            <!-- Alerts -->
            <div class="row">
                <div class="col-md-12">

                </div>
            </div>
            <!-- /Alerts -->

            <!-- Page Header -->
            <div class="page-header">
                <div class="row">
```

Figure 24 Admin Dashboard Source Code Continue

Appendices 1:3 Users Dashboard Source Codes



E:\xampp\htdocs\WORK\e-visitor\admin\edit_user.php

```php
<?php

//includes the file for user permission
include("../includes/check_login.php");
check_login();



?>

<!DOCTYPE html>
<html lang="en">

<?php @include("../includes/head.php");?>

<body>

    <!-- Sidebar -->
    <div class="sidebar" id="sidebar">
        <?php @include("../includes/sidebar.php");?>
    </div>
    <!-- /Sidebar -->

    <!-- Page Wrapper -->
    <div class="page-wrapper">

        <!-- Page Content -->
        <div class="content container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <div class="welcome-box">
<!-- <div class="welcome-img">
<img alt="" src="../assets/img/logoo.png">
</div> -->

<?php
$aid=$_SESSION['visaid'];
$sql = "SELECT *
FROM staff_tb
JOIN designation_tb ON staff_tb.designation_id = designation_tb.id
JOIN office_branch_tb ON staff_tb.office_branch_id = office_branch_tb.id
WHERE staff_tb.id=:aid";
                    //$sql="SELECT * from  staff_tb where id=:aid";
$query = $dbh -> prepare($sql);
$query->bindParam(':aid',$aid,PDO::PARAM_STR);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
```

Figure 25 Users Dashboard Source Code

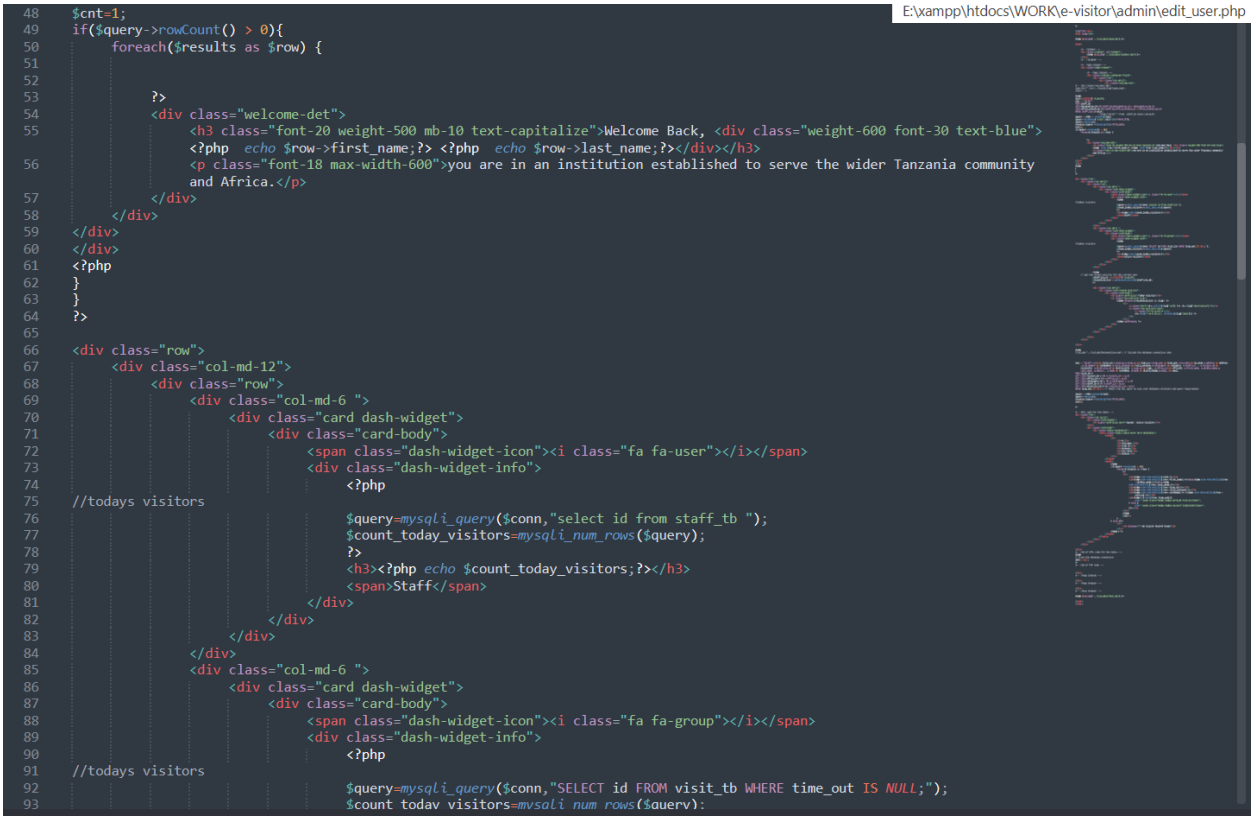


Figure 26 Users Dashboard Source Code Continue

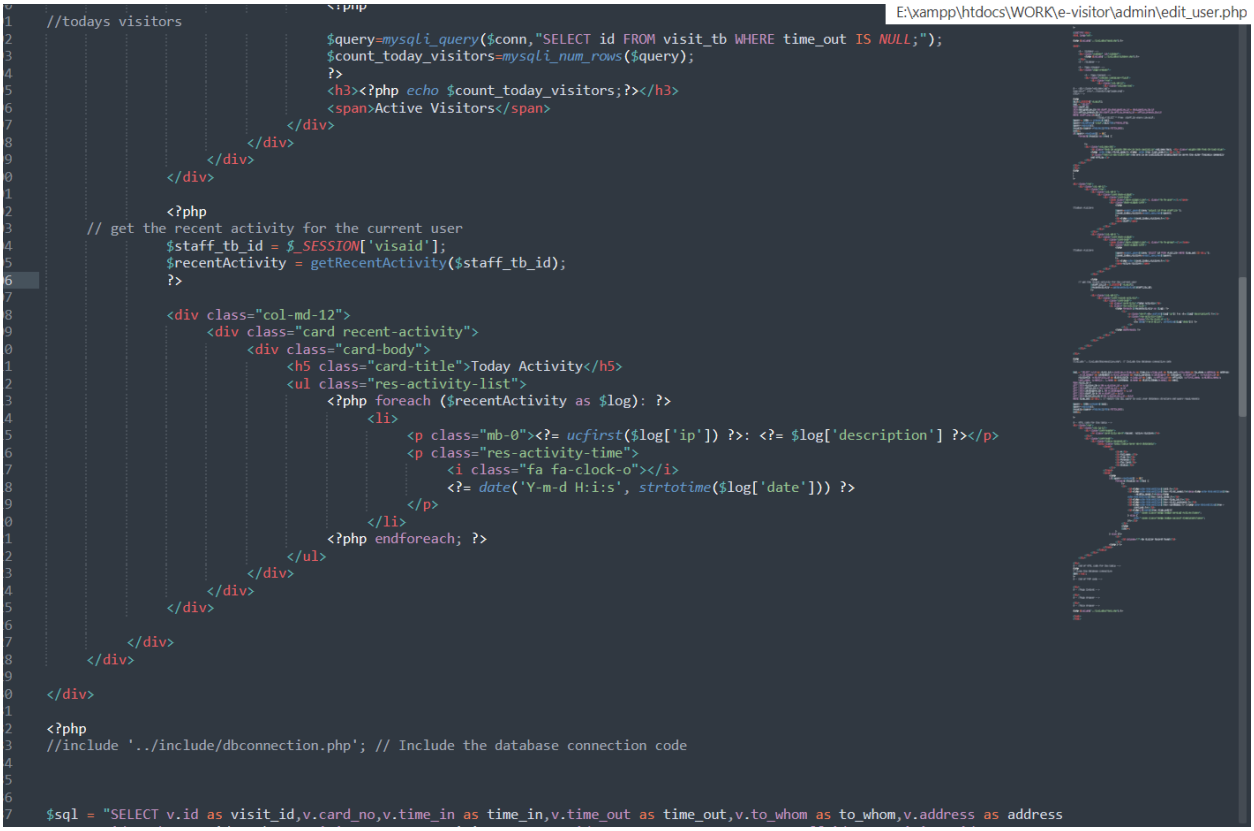
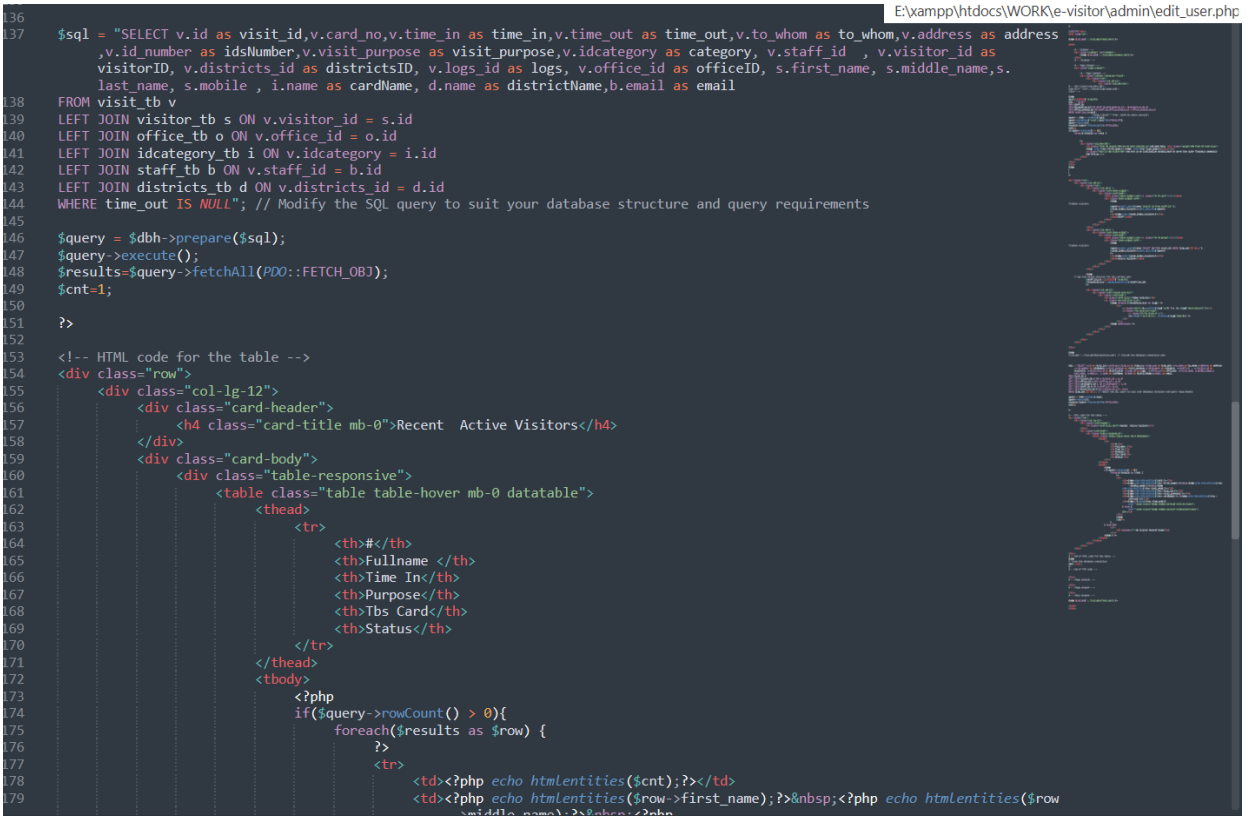
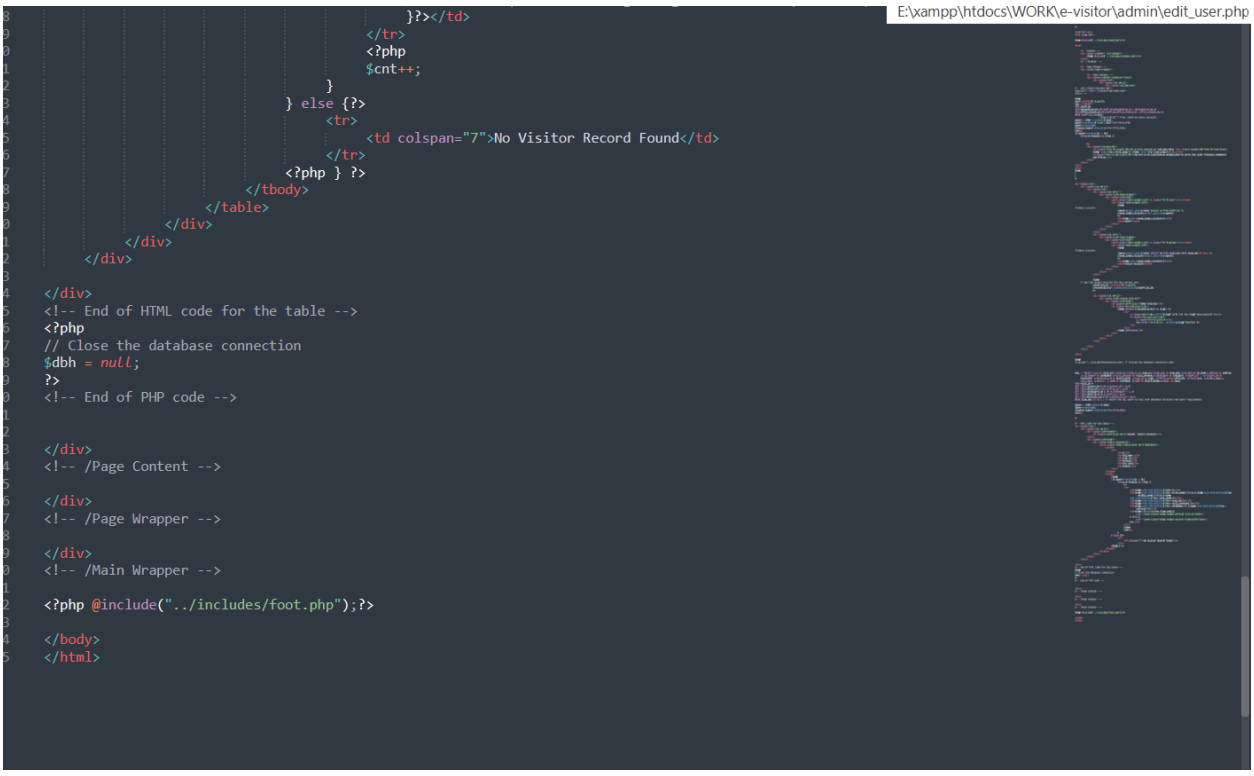


Figure 27 Users Dashboard Source Code Continue




```
$sql = "SELECT v.id as visit_id,v.card_no,v.time_in as time_in,v.time_out as time_out,v.to_whom as to_whom,v.address as address
     ,v.id_number as idsNumber,v.visit_purpose as visit_purpose,v.idcategory as category, v.staff_id  , v.visitor_id as
     visitorID, v.districts_id as districtsID, v.logs_id as logs, v.office_id as officeID, s.first_name, s.middle_name,s.
     last_name, s.mobile , i.name as cardName, d.name as districtName,b.email as email
FROM visit_tb v
LEFT JOIN visitor_tb s ON v.visitor_id = s.id
LEFT JOIN office_tb o ON v.office_id = o.id
LEFT JOIN idcategory_tb i ON v.idcategory = i.id
LEFT JOIN staff_tb b ON v.staff_id = b.id
LEFT JOIN districts_tb d ON v.districts_id = d.id
WHERE time_out IS NULL"; // Modify the SQL query to suit your database structure and query requirements

$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;

?>

<!-- HTML code for the table -->
<div class="row">
     <div class="col-lg-12">
          <div class="card-header">
               <h4 class="card-title mb-0">Recent  Active Visitors</h4>
          </div>
          <div class="card-body">
               <div class="table-responsive">
                    <table class="table table-hover mb-0 datatable">
                         <thead>
                              <tr>
                                   <th>#</th>
                                   <th>Fullname </th>
                                   <th>Time In</th>
                                   <th>Purpose</th>
                                   <th>Tbs Card</th>
                                   <th>Status</th>
                              </tr>
                         </thead>
                         <tbody>
                              <?php
                              if($query->rowCount() > 0){
                                   foreach($results as $row) {
                                        ?>
                                        <tr>
                                             <td><?php echo htmlentities($cnt);?></td>
                                             <td><?php echo htmlentities($row->first_name);?> <?php echo htmlentities($row
```

Figure 28 Users Dashboard Source Code Continue




```
                                        }?></td>
                                   </tr>
                                   <?php
                                   $cnt++;
                              }
                         } else {?>
                              <tr>
                                   <td colspan="7">No Visitor Record Found</td>
                              </tr>
                         <?php } ?>
                    </tbody>
               </table>
          </div>
     </div>
</div>
<!-- End of HTML code for the table -->
<?php
// Close the database connection
$dbh = null;
?>
<!-- End of PHP code -->


</div>
<!-- /Page Content -->

</div>
<!-- /Page Wrapper -->

</div>
<!-- /Main Wrapper -->

<?php @include("../includes/foot.php");?>

</body>
</html>
```

Figure 29 Users Dashboard Source Code Continue