Design and Implementation of Passport Consultancy Management System (PCMS)

by

Shaharia Bhuiyan Emon
ID: CSE1901016029

Suraiya Afrin Jhumur
ID: CSE1803015090

Emdadul Haque Milon
ID: CSE1901016121

MD. Anisur Rahman
ID: CSE1901016086

Supervised by
Md. Zahidul Hasan

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

January 2023

Design and Implementation of Passport Consultancy Management System (PCMS)

by

Shaharia Bhuiyan Emon
ID: CSE1901016029

Suraiya Afrin Jhumur
ID: CSE1803015090

Emdadul Haque Milon
ID: CSE1901016121

MD. Anisur Rahman
ID: CSE1901016086

Supervised by
Md. Zahidul Hasan

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

January 2023

APPROVAL

The project titled "**Design and Implementation of Passport Consultancy Management System (PCMS)**" submitted by Shaharia Bhuiyan Emon (CSE1901016029), Suraiya Afrin Jhumur (CSE1803015090), Emdadul Haque Milon (CSE1901016121) and MD. Anisur Rahman (CSE1901016086) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Md. Zahidul Hasan
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Md. Zahidul Hasan, Lecturer,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Md. Zahidul Hasan)
Supervisor

Signature

Shaharia Bhuiyan Emon
ID: CSE1901016029

Suraiya Afrin Jhumur
ID: CSE1803015090

Emdadul Haque Milon
ID: CSE1901016121

MD. Anisur Rahman
ID: CSE1901016086

ABSTRACT

Now days the world has come to hand through advanced technology. Keeping this in mind, we thought to build a system that would make the passport-related content one step easier. Our system will help the user make the work easier. It will save the time of the user. They don't have to face any hassles to go to the embassy. The user can complete visa stamping within a few clicks. Some unscrupulous brokers charge more money from the users. But using this system, users can check the correct information. They don't have to pay any extra money. Sometimes more money was taken from the head office and from the agency, the follow up of the software will not be possible, everyone can see the fixed amount from everyone's panel.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Md. Zahidul Hasan**, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Dr. Md. Alamgir Hossain,** Dean, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to our parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

CSS	Cascading Style Sheets
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
HTML	Hypertext Markup Language
IT	Information Technology
JS	Java Script
PCMS	Passport Consultancy Management System
RDBMS	Relational Database Management System
RDM	Relational Data Model
SQL	Structured Query Language

TABLE OF CONTENTS

Title			Page No.
DECLARATION ……..……..			iii
ABSTRACT ……...…….			iv
ACKNOWLEDGEMENT …….……			v
LIST OF ABBREVIATION……...			vi
CHAPTER 1			01
INTRODUCTION TO PASSPORT CONSULTANCY MANAGEMENT SYSTEM			
	1.1	Introduction…… …	01
	1.2	Objectives…….....	01
	1.3	Motivation……. .	01
CHAPTER 2			02 – 05
SYSTEM PROCESS			
	2.1	Objectives……	02
	2.2	Software Process Model……..	02
		2.2.1 Agile process model…….....	02
		2.2.2 Scrum……...	04
		2.2.3 The Product Backlog……	05
CHAPTER 3			06 –07
REQUIREMENTS			
	3.1	System Requirements……	06
		3.1.1 Initial Investigation of Candidate System…….	06

		3.1.2 Functional Requirements……..	06
		3.1.3 Non-Functional Requirements……..	07
CHAPTER 4			08 – 14
SYSTEM DESIGN			
	4.1	Class Diagram……	08
		4.1.1 ERD (Entity Relation Diagram)……	09
		4.1.2 Data Flow Diagram (DFD)……	10
	4.2	Design Tools…….…….	11
		4.2.1 CodeIgniter……..	11
		4.2.2 PHP……..	11
		4.2.3 RDBMS……	11
		4.2.4 MySQL……	13
		4.2.5 XAMPP……	13
		4.2.6 Apache…….	13
		4.2.7 Bootstrap 4 and Material Design…….	13
		4.2.8 JavaScript……	13
		4.2.9 Html…….	14
		4.2.10 CSS……	14
CHAPTER 5			15 – 24
SYSTEM IMPLEMENTATIONS			
	5.1	Public Site ……..……	15
		5.1.1 Screenshot of Home Page……	15
		5.1.2 Screenshot of Contact Page……..	15
		5.1.3 Screenshot of Login Page ……	16
		5.1.4 Screenshot of Register Page ……..……...	16
	5.2	Head Office Panel	17

		5.2.1	Screenshot of Head Office Panel Dashboard……..	17
		5.2.2	Screenshot of Add passport Type……...	17
		5.2.3	Screenshot of Modify Passport……...	18
		5.2.4	Screenshot of Consultancy……..	18
		5.2.5	Screenshot of Received List…….....	19
		5.2.6	Screenshot of Return Passport…….....	20
	5.3	Company Panel		21
		5.3.1	Screenshot of Dashboard ……..	21
		5.3.2	Screenshot of Send Passport and All Passport List……	22
		5.3.3	Screenshot of Modify Send Passport……..	23
		5.3.4	Screenshot of Return Passport…….....	23
		5.3.5	Screenshot of Reward…….	24
		5.3.6	Screenshot of Message……	24
CHAPTER 6				25
CONCLUSION AND FUTURE WORKS				
	6.1	Conclusion ……..……..		25
	6.2	Future works …….……		25
REFERENCES				26

LIST OF FIGURES

Figure No.	**Title**	**Page No.**
Fig 3.1	Different types of requirements	07

Fig 4.1	Class Diagram	08
Fig 4.2	Entity Relationship Diagram (ERD)	09
Fig 4.3	Data Flow Diagram (0 level DFD)	10

CHAPTER 1

INTRODUCTION TO PASSPORT CONSULTANCY MANAGEMENT SYSTEM

1.1 Introduction

By using the Passport Consultancy Management System (PCMS) we can do visa stamping easily. So that the users don't have to face any hassle to go to the embassy. It's restricted for all people to enter the embassy. But using this system users can check easily that has the book been stamped from the embassy to the Head office or not. All the problems of all agencies will be solved through this software. It would be much easier to issue emergency notices to all agencies at once. Previously, the agency could not work properly due to these small problems, and these problems will be their daily suffering, the agency's suffering will be removed instantly by the software.

1.2 Objectives

Our Passport Consultancy Management System's (PCMS) aim is to save time and make the work easier for the user. Every consultancy office must create a list of their passport word files and send it to the headquarters. Some of the main questions after sending the mail to the head office are:

- Have our emails been sent?
- Did I mail the names of two books?
- Is the passport list okay?
- Our system can easily answer the above questions and it will make the work much easier. Anyone can do the work in a short time. There's no need to call anyone to get the job done or check on anything related to passports.

1.3 Motivation

There is a purpose behind every action. Our target was to make passport-related processing easier so the users wouldn’t face any hassle. This system will save time. Make the work easier. The user doesn't have to pay more money. They will get the correct information on their own.

CHAPTER 2

SOFTWARE PROCESSES

2.1 Objectives

The objectives of this chapter are to introduce the idea of a software process- a coherent set of activities for software production. The objectives are:

- Understand the concept of a software process and a software process model.
- Understand a number of different software process models and when they might be used.
- Understand in outline, process models for software requirement engineering, software development, testing and evolution.

2.2 Software Process Model

A structured set of activities are required to develop a software system. A software process model is an abstract representation of a software system. Each process model represents a process from a particular perspective so only provide partial information about the process. There are different types of software process models. In our project we follow the Agile Model for software development.

Agile development practices have been widely adopted in a variety of organizations, yet only a few tools are available to help make the practical process of managing agile teams less painful and more successful. HP Agile Manager is a purpose-built SaaS-based agile planning tool that provides a simpler, more innovative way to manage collaborative development. Agile Manager can work for small startups and midsize teams and can scale up for bigger organizations as a cost-effective and flexible tool to apply agile techniques to improve your system development process. This book combines agile software development practices with HP's Agile Manager tool when used as a SaaS product or an on premise solution. It will explain how to master key functionalities of Agile Manager. It teaches you how to build and plan releases, create a product's backlog, and manage day-to-day R&D activity such as addressing user stories and defects and reviewing the team activity with the help of informative dashboards. While this book is not designed to teach agile system development per se, it starts out with a chapter that covers agile system development practices and specifically focuses on the Scrum agile methodology and its related concepts [7].

2.2.1 Agile process model

The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans

regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client. Agile model is illustrated in Fig 2.1.

Phases of Agile Model:
Following are the phases in the Agile model are as follows:

- ✓ Requirements gathering
- ✓ Design the requirements
- ✓ Construction/ iteration
- ✓ Testing/ Quality assurance
- ✓ Deployment
- ✓ Feedback

Fig 2.1. Agile Model

Requirements gathering:

In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

Design the requirements:

When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

Construction/ iteration:

When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

Testing:

In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

Deployment:

In this phase, the team issues a product for the user's work environment.

Feedback:

After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

2.2.2 Scrum

SCRUM is an agile development process focused primarily on ways to manage tasks in team-based development conditions.

There are three roles in it, and their responsibilities are:

Scrum Master:

The scrum can set up the master team, arrange the meeting and remove obstacles for the process

Product owner:

The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.

Scrum Team:

The team manages its work and organizes the work to complete the sprint or cycle.

2.2.3 The Product Backlog

The product backlog contains a list of items that generally describe the requirements of the product or bugs that need to be addressed. High-level backlog items such as themes and features consist of an organized set of requirements, whereas more detailed low-level items such as user stories and defects consist of the actual product backlog items that require addressing, thus completing the themes and features. To begin working on the product backlog and continue managing your release and team life cycle, let's switch back from the configuration area by clicking the gray gear icon (which is where we left off in the previous chapter)
Then navigate to the Product Backlog view using the main navigation bar. The Product Backlog view allows you to manage the high-level requirements through the Themes and Features sections [7].

CHAPTER 3

REQUIREMENTS

3.1 System Requirements

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems.

3.1.1 Initial Investigation of Candidate System

The first step in the system development life cycle is the identification of the need. This is a user's request to change, improve or enhance an existing system. As there is likely to be a system of such requests, standard procedures must be established to deal with them. The initial investigation is one way of handling this sort of thing. The objective is to determine whether the request is valid and feasible before a recommendation is reached to do nothing to improve or modify the existing system or build a new one.

There are three key strategies or general approach for eliciting information regarding the user requirement are:

- ✓ Asking
- ✓ Getting information from the existing information system
- ✓ Prototyping

Requirements are generally split into two types: Functional and Non-functional requirements.

3.1.2 Functional Requirements

The functional requirements for a system describe what the system should do. These requirements depend on the type of system being developed, the expected users of the system, and the general approach taken by the organization when writing requirements. When expressed as user requirements, functional requirements are usually described in an abstract way that can be understood by system users. However, more specific functional system requirements describe the system functions, its inputs and outputs, exceptions, etc., in detail. Functional system requirements vary from general requirements covering what the system should do to very specific requirements reflecting local ways of working or an organization's existing systems [4].

Some examples of functional requirements include:

- ✓ Specifications of what the system must do
- ✓ Business rules that must be met
- ✓ Steps that the system must take in authentication
- ✓ Details of what must be tracked in the system
- ✓ The reporting requirements of the system
- ✓ Specifics relating to legal or regulatory compliance
- ✓ Outlines the levels of user and their authorization
- ✓ Details of how transactions must occur
- ✓ The external interfaces of the system

3.1.3 Non-Functional Requirements

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system implementation such as the

capabilities of devices or the data representations used in interfaces with other systems. Non-functional requirements, such as performance, security, or availability, usually specify or constrain characteristics of the system as a whole. Non-functional requirements are often more critical than individual functional requirements. System users can usually find ways to work around a system function that doesn't really meet their needs. However, failing to meet a non-functional requirement can mean that the whole system is unusable. For example, if an aircraft system does not meet its reliability requirements, it will not be certified as safe for operation

If an embedded control system fails to meet its performance requirements, the control functions will not operate correctly [4].

Some examples of functional requirements include:

- ✓ The time taken for a specific page to load.
- ✓ The speed within which certain requests must be processed.
- ✓ The level of availability the system should have.
- ✓ Which functions can be performed at different times, and when maintenance will be carried out.
- ✓ How many users the system can handle concurrently.

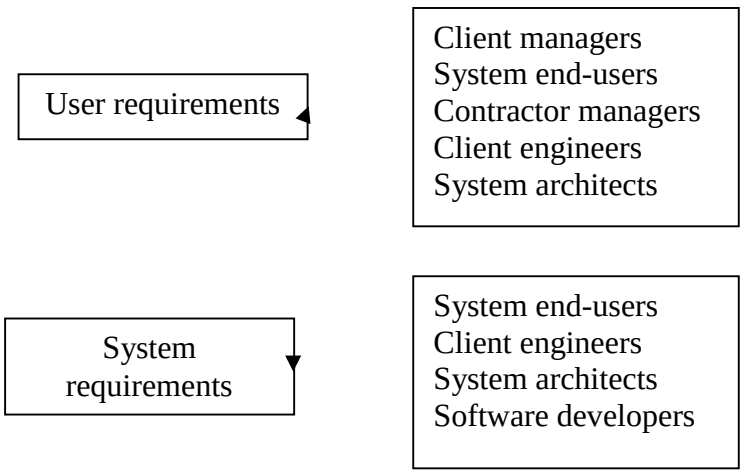


Fig 3.1. Different types of requirements

CHAPTER 4

SYSTEM DESIGN

4.1 Class Diagram

A class diagram describes the static structure of a system. It shows how a system is structured rather than how it behaves. The static structure of a system comprises a number of class diagrams and their dependencies. The main constituents of a class diagram are

classes and their relationships: generalization, aggregation, association, and various kinds of dependencies [6].

- ❖ Classes
- ❖ Attributes
- ❖ Operation
- ❖ Association
- ❖ Aggregation
- ❖ Composition

Fig: 4.1 Class Diagram

4.1.1 ERD (Entity Relation Diagram)

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses data modeling techniques that can help define business processes and serve as the foundation for a relational database.

Fig: 4.2. ERD (Entity Relation Diagram)

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later [5].

4.1.2 Data Flow Diagram (DFD)

Data flow diagram is a graphical representation of data flow in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system [6].

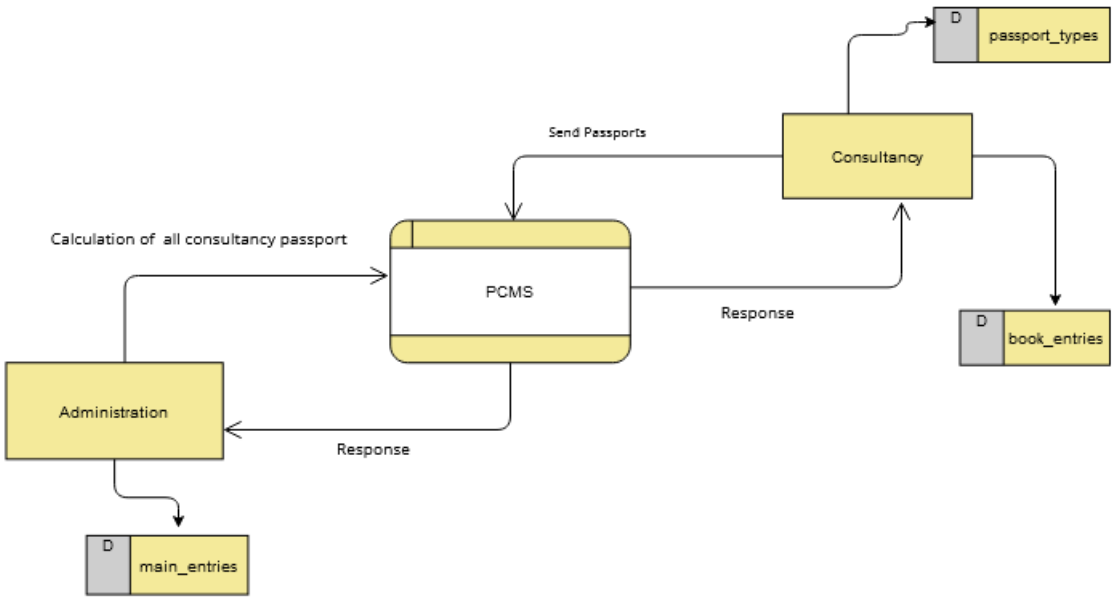


Fig: 4.3 Level DFD

DFD Components DFD can represent Source, destination, storage and flow of data using the following set of components

- Entities - Entities are source and destination of information data. Entities are represented by rectangles with their respective names.
- Process - Activities and action taken on the data are represented by Circle or Round edged rectangles.
- Data Storage - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
- Data Flow - Movement of data is shown by pointed arrows. Data movement is shown from the base of the arrow as its source towards the head of the arrow as destination [6].

4.2 Design Tools

The system is expected to serve all the purpose that it has been developed for. We use the following tools to implement this system:

Backend Framework:

- ✓ CodeIgniter 04
- ✓ PHP-7.4.3
- ✓ RDBMS: MySQL
- ✓ Server: XAMPP, Apache

Fronted framework:

- ✓ BOOTSTRAP
- ✓ JAVASCRIPT
- ✓ HTML
- ✓ CSS

4.2.1 CodeIgniter

CodeIgniter is a high performance PHP framework for developing MVC-based web applications. CodeIgniter contains libraries, simple interface and logical structure to access these libraries, plug-ins, helpers and some other resources which solve the complex functions of PHP more easily maintaining a high performance. It simplifies the PHP code and brings out a fully interactive, dynamic website at a much shorter time. It supports PHP version 5.2.6 or newer and MySQL version 4.1 or newer. It makes your web more robust and your code easier to read and maintain. It is a free toolkit, light weight and easier to install.

4.2.2 PHP

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications.

4.2.3 RDBMS

A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. RDBMS stores data in the form of tables, with most commercial relational database management systems using Structured Query Language (SQL) to access the database. However, since SQL was invented after the initial development of the relational model, it is not necessary for RDBMS use. The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering. Combination of system performance and ease of implementation [8].

Relational Data Model (RDM)

The relational model uses a collection of tables to represent both data and the relationships among those data. Tables are logical structures maintained by the database manager. The relational model is a combination of three components, such as

- Structural,
- Integrity and
- Manipulative parts.

- **Structural Data**

  The structural part defines the database as a collection of relations.

- **Integrity**

  The database integrity is maintained in the relational model using primary and foreign keys.

- **Manipulative parts**

  The relational algebra and relational calculus are the tools used to manipulate data in the database. Thus relational models have a strong mathematical background.

The key features of relational data model are as follows:

– Each row in the table is called tuple.

– Each column in the table is called an attribute.

– The intersection of the row with the column will have data value.

– In relational model rows can be in any order.

– In relational model attributes can be in any order.

– By definition, all rows in a relation are distinct. No two rows can be exactly the same.

– Relations must have a key. Keys can be a set of attributes.

– For each column of a table there is a set of possible values called its domain.

– Domain is the set of valid values for an attribute.

– Degree of the relation is the number of attributes (columns) in the relation.

– Cardinality of the relation is the number of tuples (rows) in the relation.

Relational Algebra:

The relational algebra is a theoretical language with operations that work on one or more relations to define another relation without changing the original relation. Thus, both the operands and the results are relations

Hence the output from one operation can become the input to another operation. This allows expressions to be nested in relational algebra. This property is called closure. Relational algebra is an abstract language, which means that the queries formulated in relational algebra are not intended to be executed on a computer. Relational algebra consists of a group of relational operators that can be used to manipulate relations to obtain a desired result. Knowledge about relational algebra allows us to understand query execution and optimization in relational database management systems.

4.2.4 MySQL

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

4.2.5 XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.

4.2.6 Apache

It is an HTTP cross-platform web server. It is used worldwide for delivering web content. The server application has been made free for installation and used for the community of developers under the aegis of Apache system Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.

4.2.7 Bootstrap 4 and Material Design

Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other 6 interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

4.2.8 JavaScript

JavaScript (JS) often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it [1].

4.2.9 Html

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. [4] Web browsers receive

HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items [3].

4.2.10 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content [3].

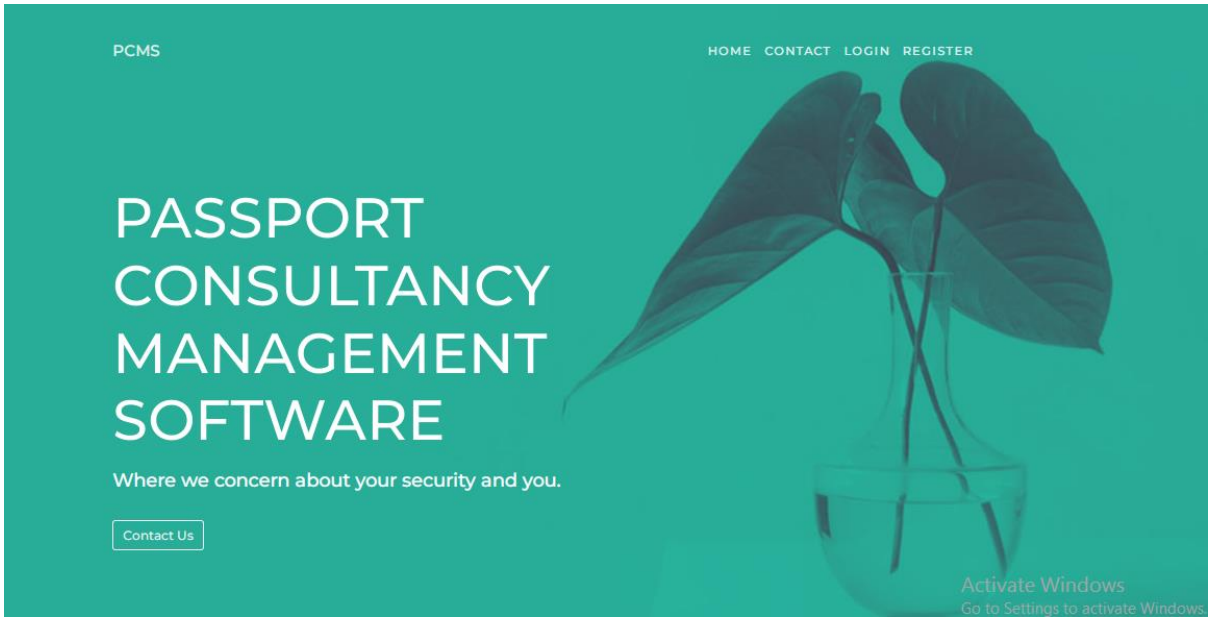
CHAPTER 5

SYSTEM IMPLEMENTATIONS

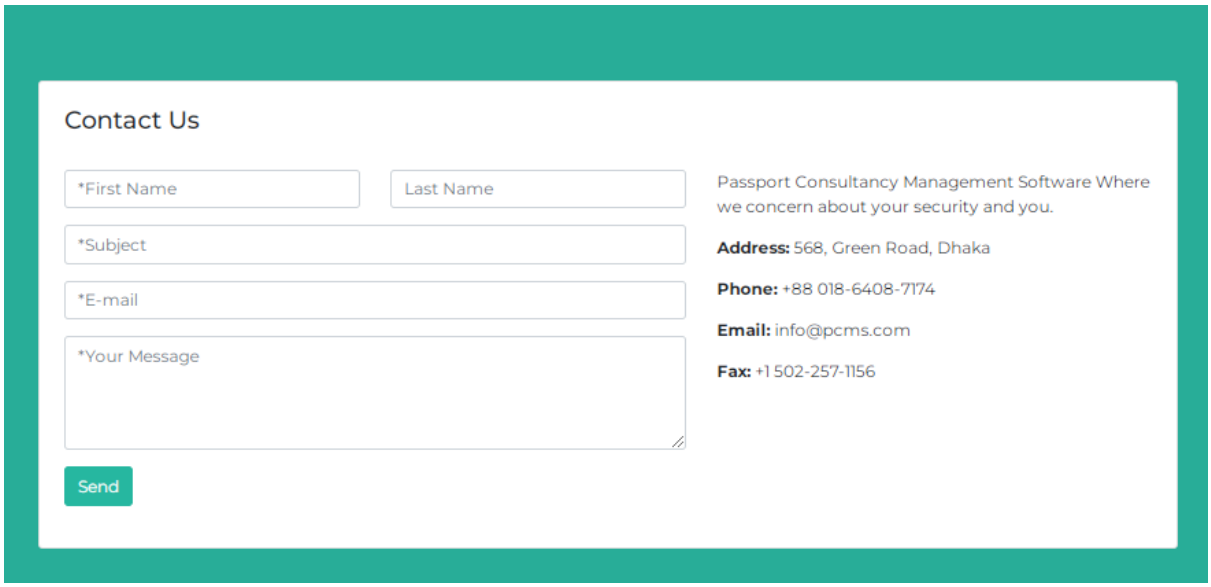
5.1 Public site

A location on the World Wide Web that is accessible by anyone with a Web browser and access to the Internet.

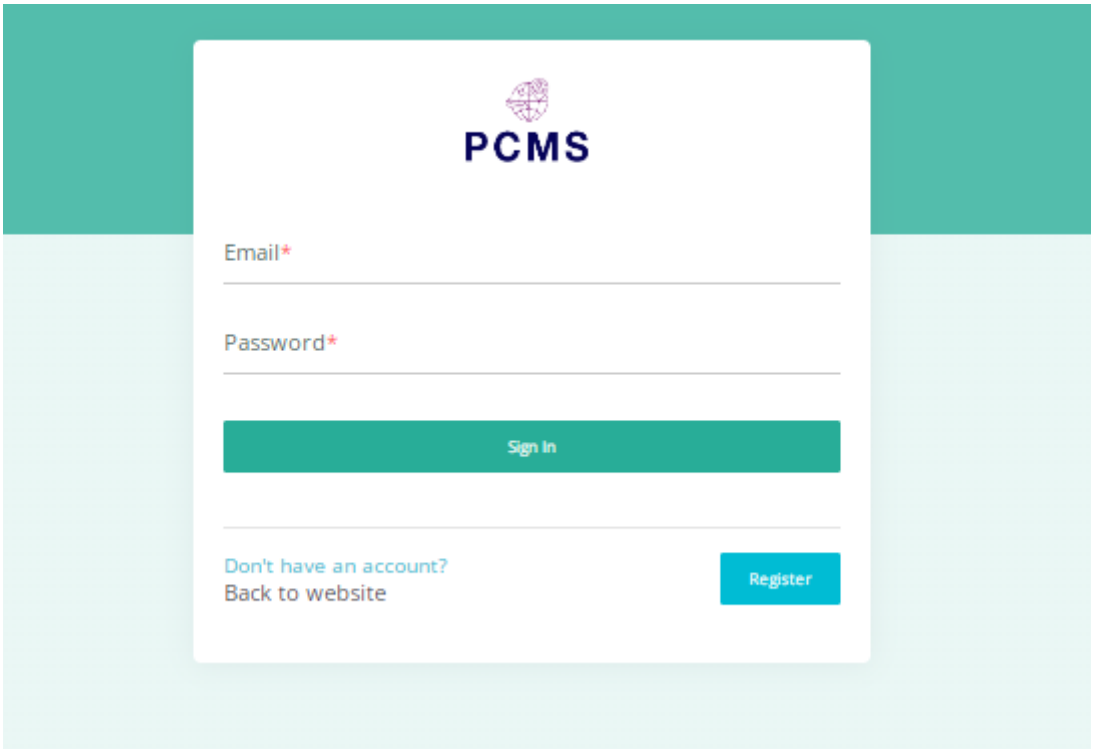
5.1.1 Screenshot of Home Page



5.1.2 Screenshot of Contact Page



5.1.3 Screenshot of Login Page

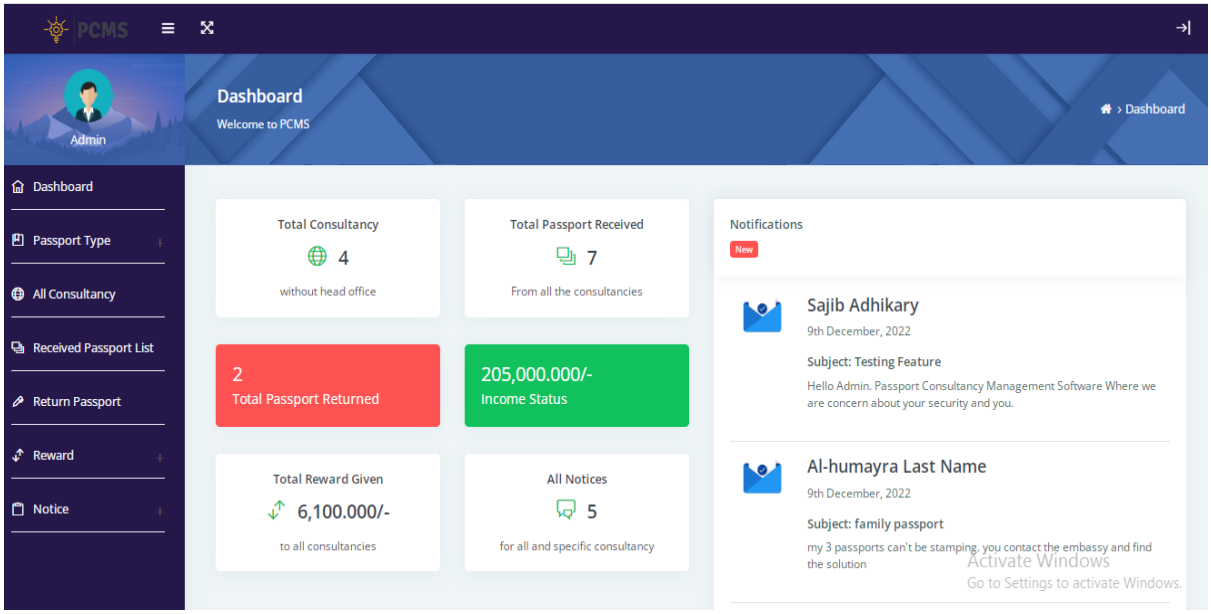


5.1.4 Screenshot of Register Page

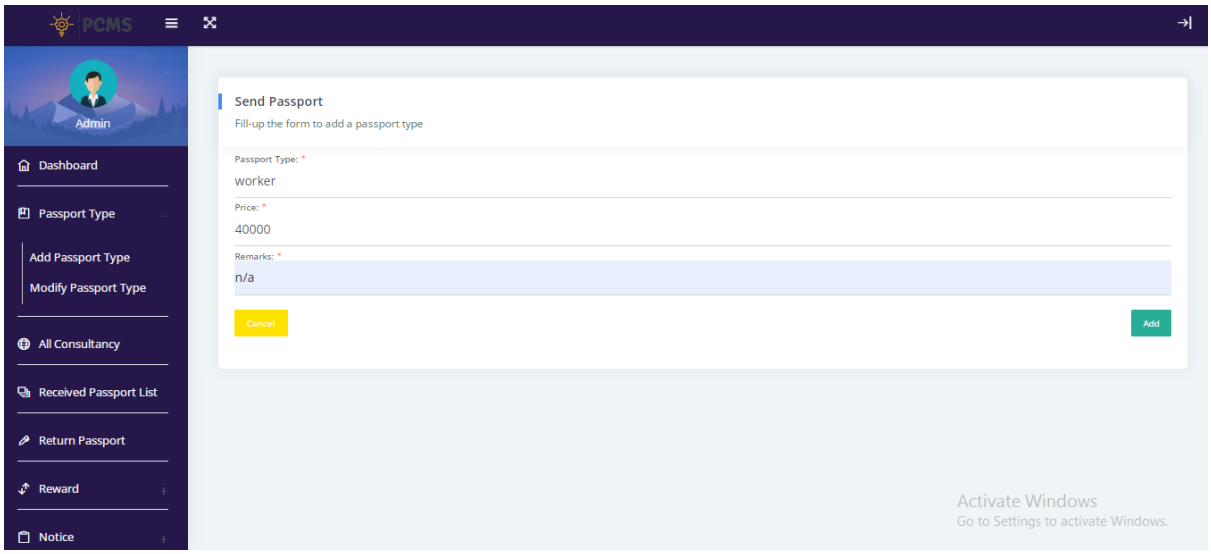
5.2 Head office Panel

Head office is a private section no user can enter. The office that maintains the system can get access. The head office will control the user (company) offices.

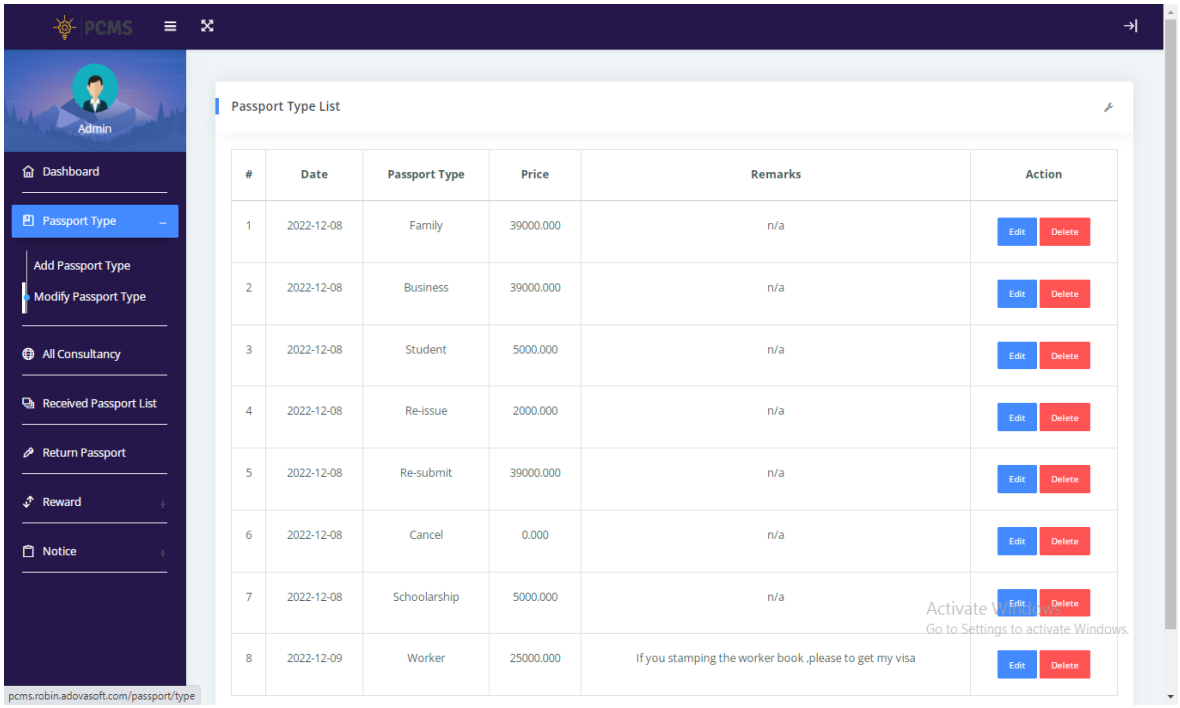
5.2.1 Screenshot of Head Office Panel Dashboard



5.2.2 Screenshot of Add Passport Type

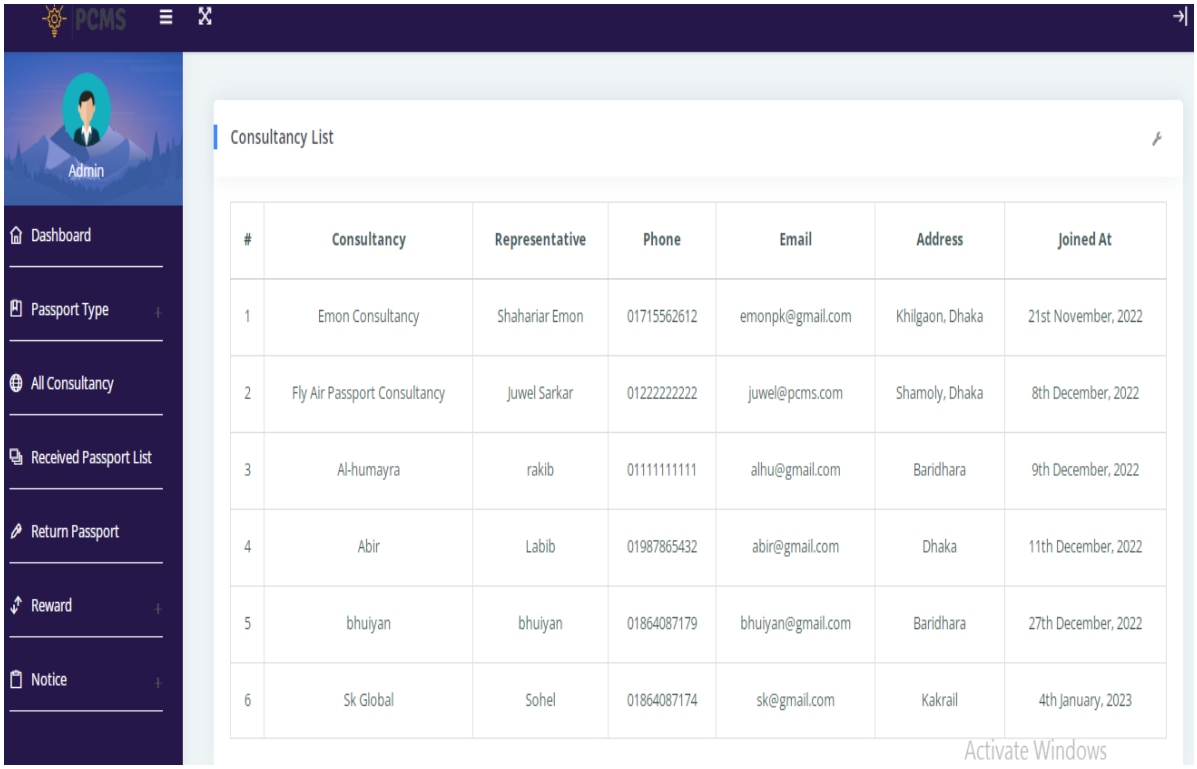


5.2.3 Screenshot of Modify Passport



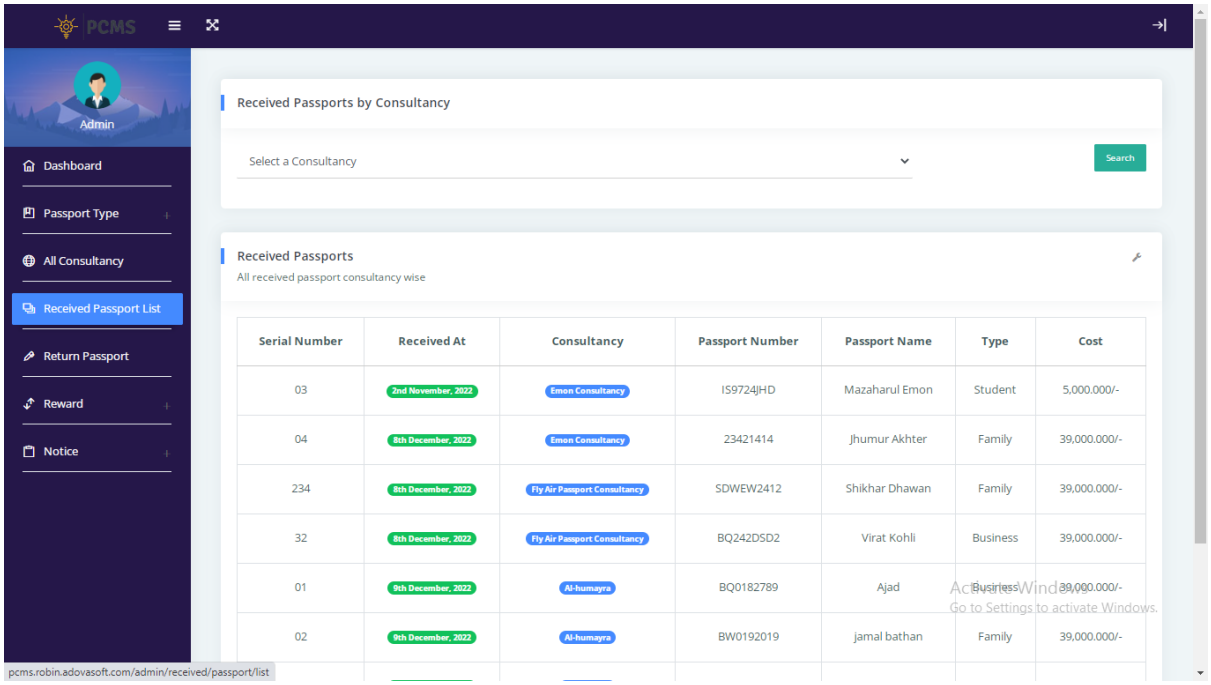
5.2.4 Screenshot of Consultancy

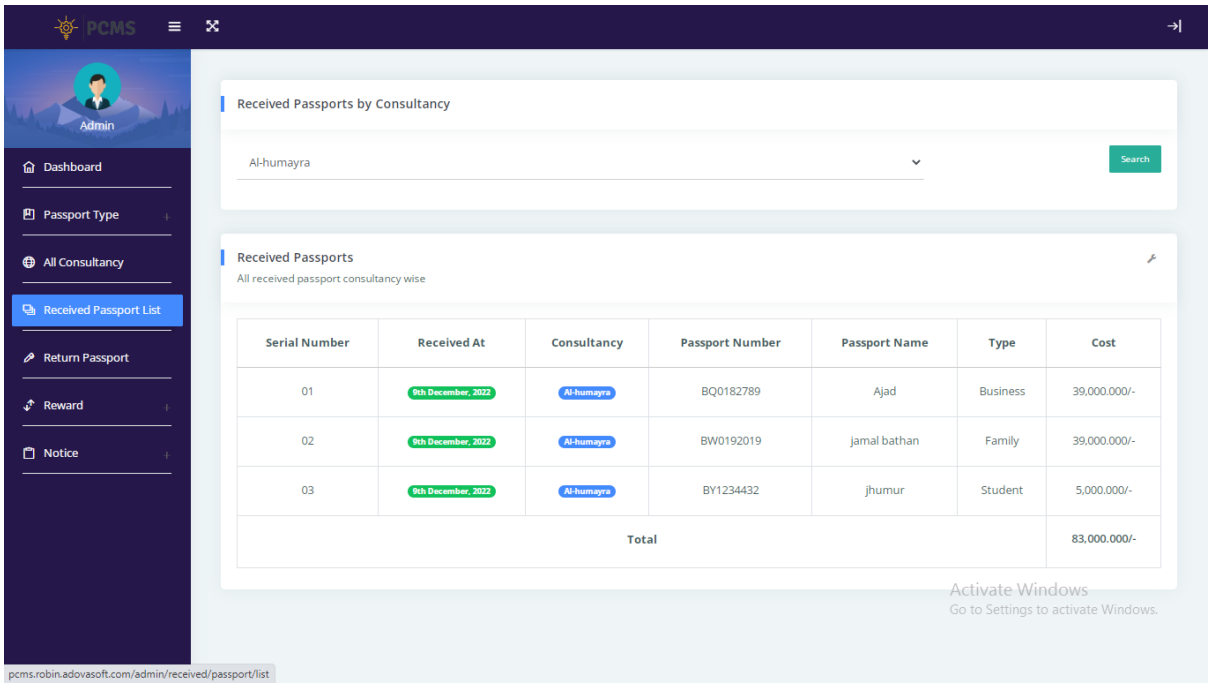
Users who register will have their own details Store.



5.2.5 Screenshot of Received List

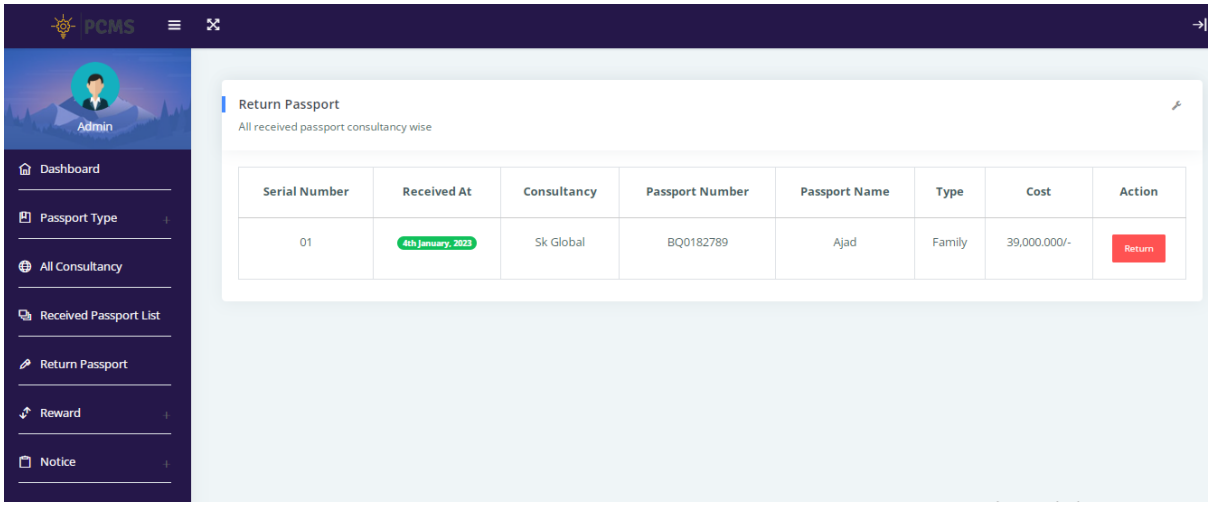
Head office can check all the passports given by the user and individual user's submitted passports.

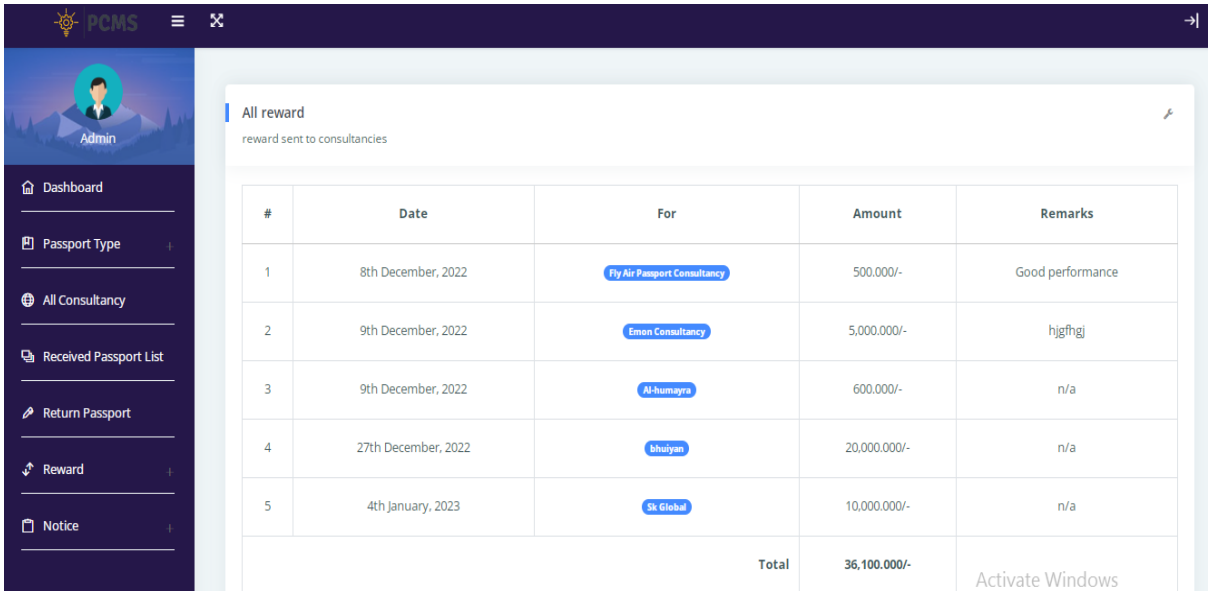




5.2.6 Screenshot of Return Passport

The passports of those whose passports do not get visa stamping are returned from the head office, And the highest passport donor company is rewarded.

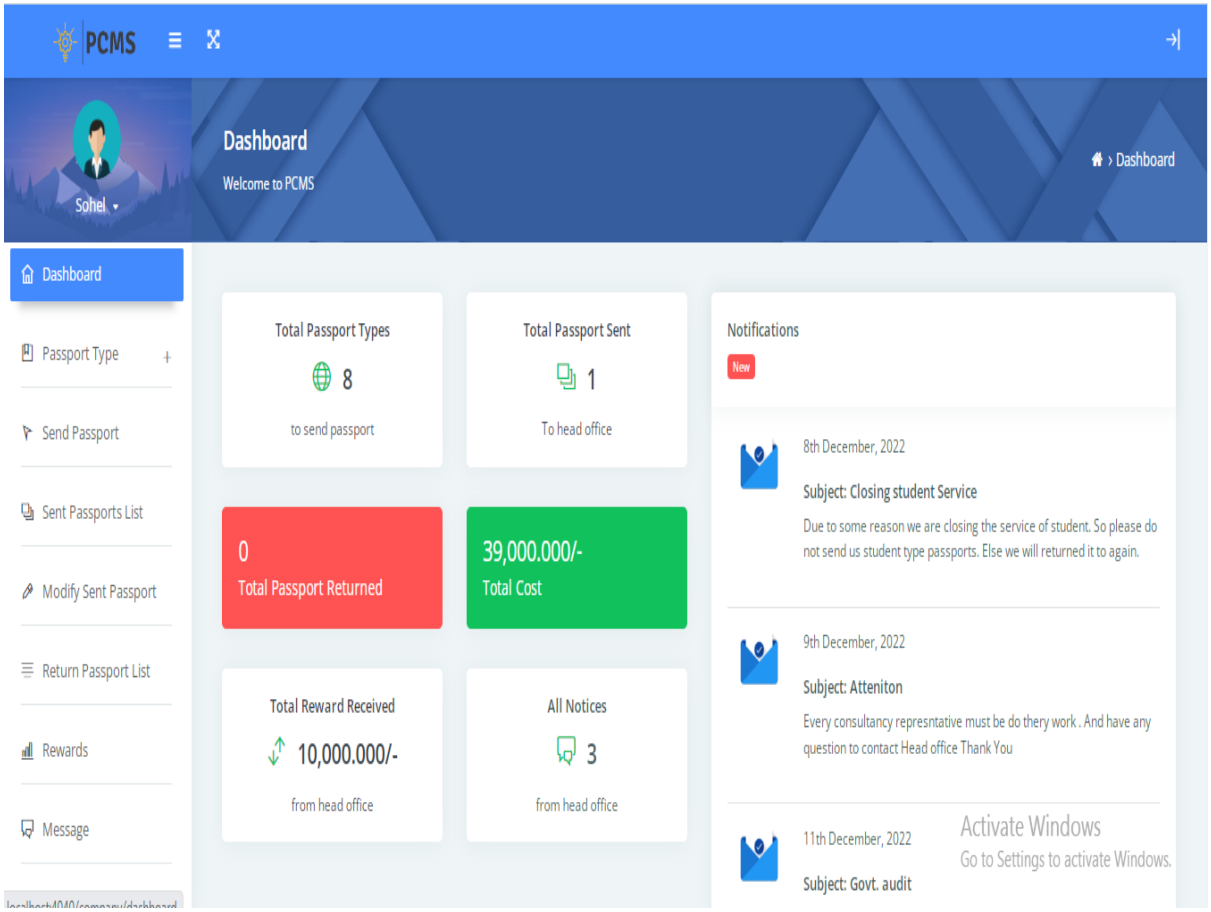




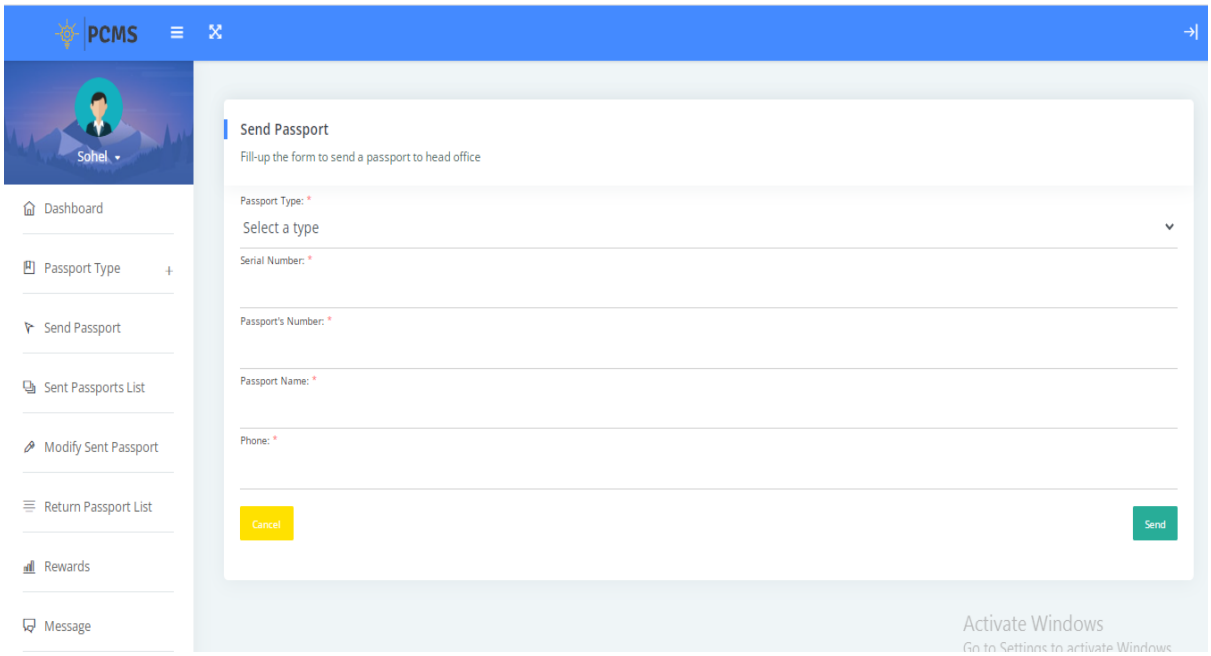
5.3 Company Panel

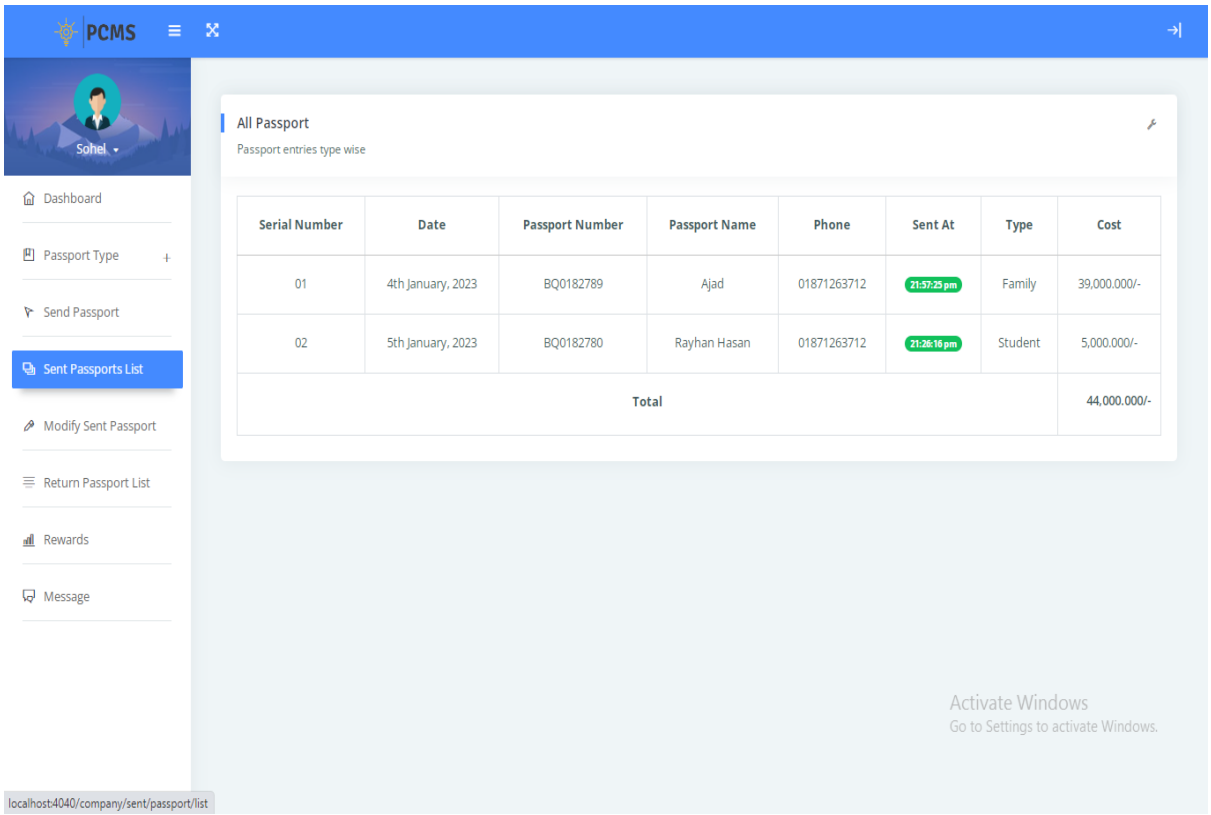
Company panel is all system users. Users' Passport entry can be modified and if they have any question they can tell it through a message. Their reward message can also be known through the system.

5.3.1 Screenshot of Dashboard



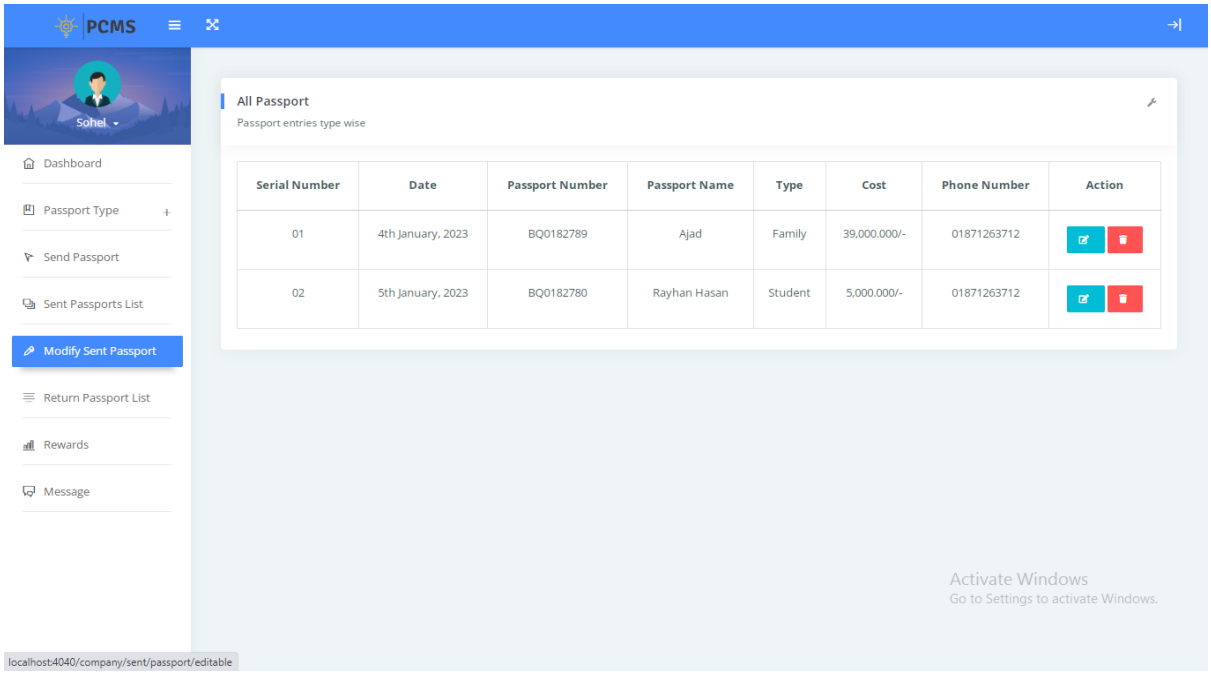
5.3.2 Screenshot of Send Passport and All Passport List



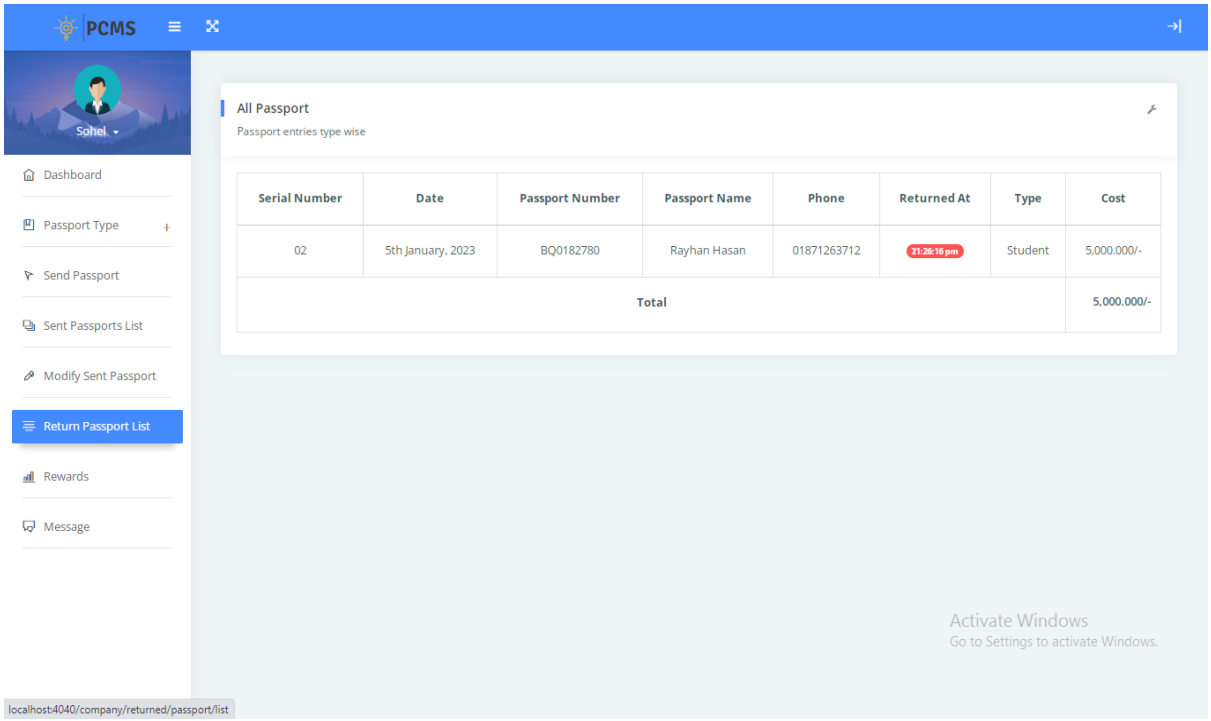


5.3.3 Screenshot of Modify Send Passport

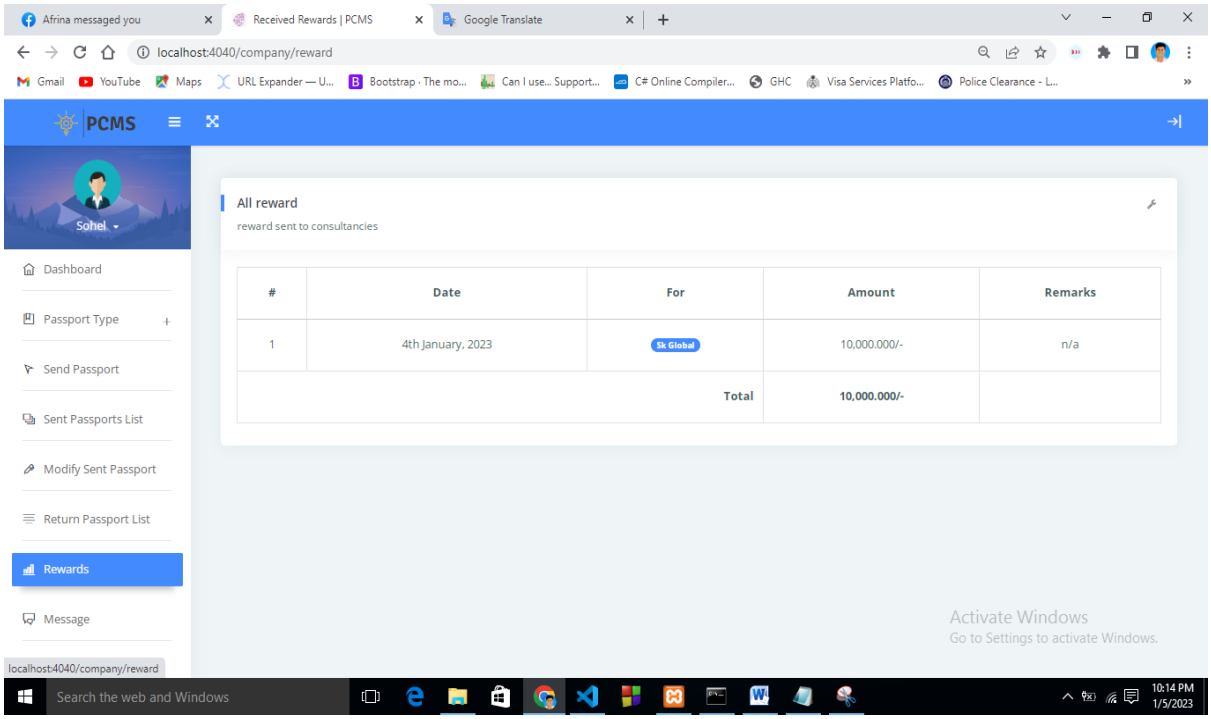
After entering in the passport system, if someone has typed wrong information, it is easy to edit from the company panel.



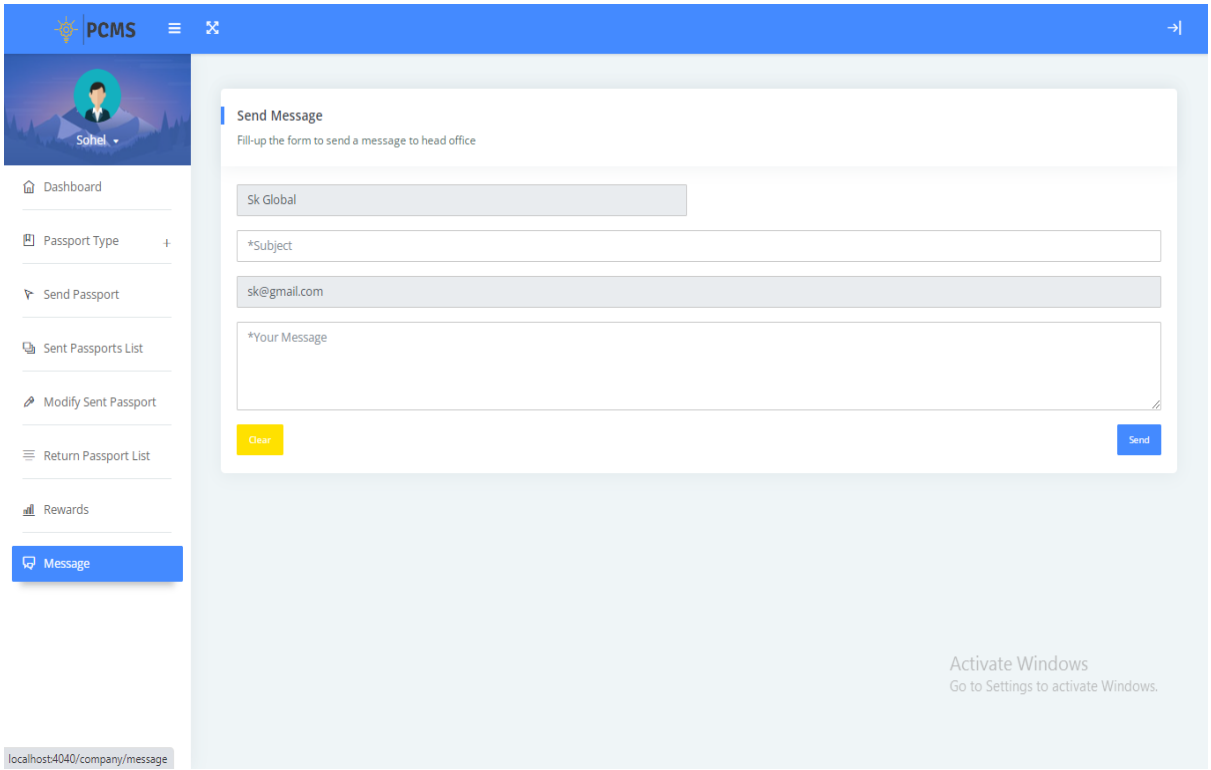
5.3.4 Screenshot of Return Passport



5.3.5 Screenshot of Reward



5.3.6 Screenshot of Message



CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this project, we have designed and implemented a system product that will make passport-related content one step easier. Visa stamping will be way easier for the users. They don't have to face any kind of hassle to go to the passport office. The head office can't take more money from the agency. It'll save time and money.

6.2 Future Works

Although we can add and delete a few things in this system, it is not fully customized. So, we are working to improve the system for customized use.

In the future we will add the following features:

- ✓ We will make this system more secure and provide a more user-friendly interface.
- ✓ We will send important documents and all notices through email and mobile SMS.
- ✓ Payments through an Online banking system will be introduced.
- ✓ All consultancy reward money will go to their bank account through the system.

REFERENCES

[1] Haverbeke,M. Eloquent,"JavaScrip A Modern Introduction to Programming", (2nd edition), (2014), No Starch Press, pp. 9-12.

[2] Kromann, F. "Beginning PHP and MySQL: From Novice to Professional" (5th edition), (2018), Apress, pp. 105-120.

[3] Fielding, J."Beginning Responsive Web Design with Html5 and Css3", (2014), Apress, pp. 75-93.

[4] Sommerville, I. "System Engineering", (10th edition),(2015) ,Pearson, pp.105-110.

[5] Al-Masree, H.K. "Extracting Entity Relationship Diagram (ERD) From Relational Database Schema", International Journal of Database Theory and Application, vol. 8, no 3, (2015), Available at: https://doi.org/10.14257/ijdta.2015.8.3.02

[6] Juristo, N. and Moreno, A. "Lecture Notes on Empirical System Engineering (System Engineering and Knowledge Engineering)", World Scientific Publishing Company, (1st edition), (2003), pp. 78-112.

[7] Tal, L,"Agile System Development with HP Agile Manager", (1st edition), (2015), Apress, pp. 1-53.

[8] Sumathi and Esakkirajan, "Fundamentals of Relational Database Management Systems", (Studies in Computational Intelligence, 47). (2007th edition), (2007) Springer, pp. 120-134.