

**WEB BASED CUSTOMER PROJECT AND TASK MANAGEMENT SYSTEM**

**BY**

**MD. ASHRAFUZZAMAN**

**ID: 122-15-1973**

This Report Presented in Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

**Nazmun Nessa Moon**

Associate Professor

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**JANUARY 2022**

## APPROVAL

This Project titled “**Web Based Customer Project and Task Management System**”, submitted by **MD. ASHRAFUZZAMAN, ID No: 122-15-1973** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **04 January 2022**.

### BOARD OF EXAMINERS



---

**Dr. Touhid Bhuiyan**  
**Professor and Head**

**Chairman**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



---

**Dr. Fizar Ahmed**  
**Assistant professor**

**Internal Examiner**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



---

**Nusrat Jahan**  
**Senior Lecturer**

**Internal Examiner**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



---

**Dr. Mohammad Shorif Uddin**  
**Professor**

**External Examiner**

Department of Computer Science and Engineering  
Jahangirnagar University

## DECLARATION

We hereby declare that this project has been done by us under the supervision of **Nazmun Nessa Moon, Associate Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

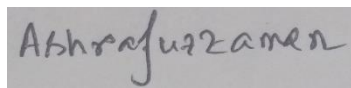
### Supervised by:



---

**Nazmun Nessa Moon**  
Associate Professor  
Department of CSE  
Daffodil International University

### Submitted by:



---

**Md. Ashrafuzzaman**  
ID: 122-15-1973  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

Firstly, I am expressing my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the final year project/internship successfully.

I am really grateful and wish my profound indebtedness to **Nazmun Nessa Moon, Associate Professor**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*web based*” to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express our heartiest gratitude to Dr. Touhid Bhuiyan, Professor and Head, Department of CSE, for his kind help to finish my project and also to other faculty members and the staff of CSE department of Daffodil International University.

I would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

## **ABSTRACT**

This project “**Web Based Customer Project and Task Management System**” is a web application by which any company can manage tasks among its employees. This project has various small parts like commenting on tasks, uploading and downloading files, task assignment, editing existing and creating new projects, task, employee, user etc. The task management system eliminates manual request and assignment of work. Users can request any work task online. Administrative staff can review the incoming work tasks and assign them to appropriate personnel. Development process of the system starts with System analysis. To develop this project, I have used MySQL database, PHP (Laravel FrameWork), HTML, CSS(Bootstrap), JQuery as programming language and PhpStorm, Visual Studio Code, Sublime Text as Editor; Firefox and Google Chrome as testing Browser. This web site was tested in many ways after implementation and it works successfully.

## TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Figures	viii
List of Tables	x
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-3</b>
1.1 Introduction	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	3
<b>CHAPTER 2: BACKGROUND</b>	<b>4-7</b>
2.1 Introduction	4
2.2 Related Works	4

2.3 Comparative Studies	6
2.4 Scope of the Problem	7
2.5 Challenges	7
<b>CHAPTER 3: REQUIREMENT SPECIFICATION</b>	<b>8-13</b>
3.1 Business Process Modeling	8
3.2 Requirement Collection and Analysis	9
3.3 Use Case Modeling and Description	9
3.4 Logical Data Model	11
3.5 Design Requirements	13
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>14-21</b>
4.1 Front-End Design	14
4.2 Back-End-Design	18
4.3 Interaction Design and UX	20
4.4 Implementation Requirements	21
<b>CHAPTER 5: IMPLEMENTATION AND TESTING</b>	<b>22-40</b>
5.1 Implementation of Database	22
5.2 Implementation of Front-end Design	24
5.2.1 User Module	25
5.2.2 Admin Module	30
5.3 Implementation of Interactions	36
5.4 Testing Implementation	36
5.5 Test Results and Reports	40

<b>CHAPTER 6: CONCLUSION AND FUTURE SCOPE</b>	
6.1 Discussion and Conclusion	<b>41</b>
6.2 Scope for Future Developments	41
	41
<b>REFERENCES</b>	
	<b>42</b>
<b>APPENDIX</b>	
Appendix A: Project Reflection	<b>43</b>
	43
<b>PLAGIARISM</b>	
Plagiarism Report	<b>44</b>
	44



## LIST OF FIGURES

<b>FIGURES</b>	<b>PAGE NO</b>
Figure 2.1: Perfexcrm website	5
Figure 3.1: Task Process Business Process Model	8
Figure 3.2: Users Use Case Model	10
Figure 3.3: The system's ER diagram	12
Figure 4.1: Code and Design for the Customer Registration Page.	15
Figure 4.2: Customer Project Detail Page for Customer	16
Figure 4.3: Web Browser Admin Dashboard Page	17
Figure 4.4: The Home Page for the Database	18
Figure 4.5: Page of Migration	19
Figure 4.6: User Page Archiving	19
Figure 5.1: PHP (Laravel Framework) Code for Database Connection	22
Figure 5.2: Database: MySQL	23
Figure 5.3: Home Page of the User	24
Figure 5.4: Home Page for Admins	25
Figure 5.5: Login Page for Users	26
Figure 5.6: Page for User Registration	26
Figure 5.7: After you have logged in	26
Figure 5.8: Updates to User Profiles	26
Figure 5.9: Customer Project	27
Figure 5.10: Customer Support Ticket	27
Figure 5.11: Customer Invoices	28
Figure 5.12: Customer Payments	28

Figure 5.13: Customer New Ticket	29
Figure 5.14: Customer Ticket Reply	29
Figure 5.15: Admin Task List	30
Figure 5.16: Home Page of the Admin	31
Figure 5.17: Customer List Page	32
Figure 5.18: Member List of the Admin	32
Figure 5.19: Task list for Admin	33
Figure 5.20: Admin Invoice List	33
Figure 5.21: Payment List for Admins	34
Figure 5.22: List of Admin Projects	34
Figure 5.23: Admin Event Calendar	35
Figure 5.24: Timesheets for Admins	35

## **LIST OF TABLES**

### **TABLES**

### **PAGE NO**

Table 5.1: Testing Information and Date

36

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Customer project and task management is an online service which can manage customer tasks among the software company team members.

This is an automated support system which can be used in different departments for administrative work. Most of the software companies handle customers' required tasks manually by using mobile phone or text message. By managing the customer tasks request automatically, this system diminishes manual intercession.

Different kinds of users like Super Admin, Project manager, Developer and customer can use this system and they can also assign tasks. If a project manager places a task to the developer, the customers also can view it from his panel. In this project tasks will have all kinds of information like task details, estimate time and budget for the developer. Then the Project manager assigns tasks to the respective developers. After completing the assigned tasks, developers will change the project tasks status to complete. Reports are available for users to check the pending/completed project tasks, the average working time of the project tasks. Customer users can take any support project wise via the support ticket system. Customers also can pay any money via online payment gateway.

This documentation is taken after with the Technique portion which clarifies Scope of the venture, software package description, useful necessities in conjunction with client case screenshots, Information administration covers the database connections, Non-functional necessities and conclusion are laid out.

## **1.2 Motivation**

Day by day, software companies are expanding rapidly. On the other hand, majority companies do not have their own online customer project and tasks management system. In that case I decided to develop a project and task management system for customers.

In this platform customers can get the opportunity to check the project status, check task status, create new tasks, send support tickets.

In the present era, time is most valuable for all people. So that customers don't have enough time to knock manually to service holders for information. As it were, automation system tools can decrease the complexities of software companies and Clients. That's why I understand that a web-based customer project and task management system which is the best tool for customers and software companies to decrease complexities for project tasks management.

## **1.3 Objectives**

- To induce a summary of project tasks from the dashboard.
- To get full information about the project.
- To get project wise tasks and payment at any specific date range.
- To urge and presume about the project deadline and costing.
- To decrease customer burden to get information manually and run business efficiently.
- To check the project wise, assign a member timesheet.
- To save time and money.

## **1.4 Expected Outcome**

1. The company owner or site administrator will be able to control the task process system by using this website.
2. Customers will be able to check the status of their projects via the customer panel.

3. Customers will be able to check task status and create new tasks using my website from their homes, offices, or anyplace else.
4. A support ticket can be used to communicate with the admin user via the website.
5. Customers won't have to physically visit the software firm.
6. This system will save them time, effort, money, and a variety of other resources.

## **1.5 Report Layout**

### **Chapter 1: Introduction**

This chapter has examined the project's motivation, aims, and intended outcome. In the final section of this chapter, we discussed the report layout.

### **Chapter 2: Background**

My project's background circumstances have been detailed here. In comparison to comparable websites, the scope of the problem and project challenges are outlined here.

### **Chapter 3: Requirement Specification**

The prerequisites are discussed in this chapter. This chapter covers business process modeling, requirement gathering and analysis, project use case modeling with descriptions, logical relational database modeling, and design requirements.

### **Chapter 4: Design Specification**

All of the project's designs are included in this chapter. Front-end design, back-end design, interaction design, and user experience design, as well as the implementation requirements.

### **Chapter 5: Implementation and Testing**

This chapter covers database implementation, front-end designs and interactions, testing implementation, test results, and reporting.

### **Chapter 6: Conclusion and Future Scope**

The final chapter discusses the project's end and future scope.

## CHAPTER 2

### BACKGROUND

#### 2.1 Introduction

In order to begin a development project, I need to study and investigate relevant projects and prior work. For successfully finishing a project, this chapter's background is critical. Other past efforts that are related to my project can provide me with important work information that I need to complete my project. First and foremost, I need to see if this type of website has been done before. I also receive a lot of related websites from Google. Then I concentrate on their project or work's limits. I concentrated on overcoming their constraints after identifying them. I eliminate the restriction and enhance or expand the website. Finally, I'll be able to create a new website for the management system. I won't be able to come up with ideas for existing features of linked projects unless I explore similar works. Consequently, our ability to improve existing features and create new ones may be limited. As a result, the project's background is crucial to its success.

#### 2.2 Related Work

A few related websites that offer services around the world are shown here. The majority of them contain a combination of functionality such as customer management, payment management, task management, and so on. However, they did not have any task management tools. Bangladesh has access to a few platforms:

- Perfexcrm.com [1]
- Rise.fairsketch.com [2]
- Taskhub.company [3]
- Crm.infyom.com [4]
- Peopleprohrm.com [5]
- High.waptechy.com [6]

**Features:** All these websites serve as standard platforms for handling consumer tasks. Not only do they provide task management services, but they also provide accounting services.

- Only the project list is provided.
- Some have only a few functionalities.
- Can assign tasks to team members.

### Problems:

- Not any project summary list.
- Cannot show project milestones.
- It is impossible to interact directly with the company's leadership.

For Bangladesh, there are basically no own administration online task management services which are managed by the legitimate authority. Perfexcrm.com [1] is a related project management software that runs through a website like the one illustrated in Figure 2.1, although it has several restrictions.

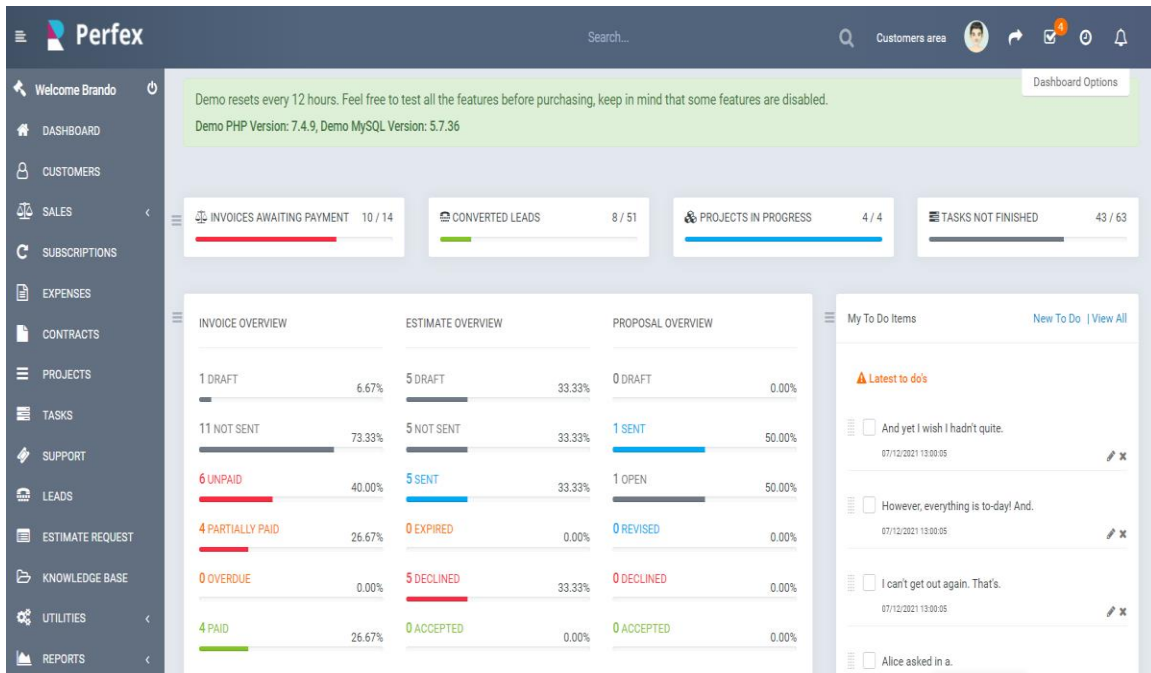


Figure 2.1: Perfexcrm website.

### Features:

- It is possible to create a project online.



- The project's information is provided.
- Information about the task is provided.
- Option for a support ticket.

**Problems:**

- They lack project management and notification features.
- In this system, there is no way to message with them.
- Users are not able to register with them.
- They put themselves at risk of security difficulties.

## **2.3 Comparative Studies**

There are numerous features on my website. When compared to the previously stated websites, some are similar and others are different. Furthermore, I created it with complete information, making it more user-friendly for users, administrators, and the authority of the software firm.

- Through my project, a customer can verify the status of a project by a certain date. However, there are some projects where a customer must conduct a manual search.
- Other projects do not allow customers to check task summary by date range or status, but I have added this feature in my project.
- Customers are welcome to provide comments on any of my projects' tasks. A new feature that compares the existing websites is task comment.
- The problem of security is crucial. For users, I used the MySQL database system, which is controlled by the company's owner.
- If a task is not confirmed, a customer can examine his task history (upcoming, current, and past) and cancel it. In comparison to other projects, this is also a novel developing characteristic of mine.

## 2.4 Scope of the problem

I've had a lot of difficulties completing this project. However, I was able to overcome these obstacles. Where do I find out about the software company's task management process? As a result, I gathered information on software companies in Bangladesh. A critical consideration is which database to use. As a database, I've decided on MySQL. Using Phpstorm as an editor, I developed my project using a variety of programming languages like LARAVEL, PHP, JAVASCRIPT (PHP Framework). For the design of this website, I must employ markup languages such as HTML and CSS. For the design of this website, I used various JAVASCRIPT library functionalities such as JQuery. This programming language was chosen because it is simple to use for developers. Because this website is user-friendly, I anticipate that many individuals will utilize it to decrease their complications.

## 2.5 Challenges

The following are some of the difficulties I've encountered:

- MySQL was a brand-new tool for me. I wanted to study a lot about this database because I needed to use it in our project.
- To come up with a gorgeous design and features that are unique.
- To create an easy-to-use interface.
- Managing two components of a website, general users and administrators.
- Safeguarding data.
- Unauthorized users are prevented from reading, writing, or removing data from the database.

# CHAPTER 3

## REQUIREMENT SPECIFICATION

### 3.1 Business Processing Model

A business process model is a flowchart or model that is an essential tool for developing a project or system. A business process model is a mapping concept that describes how data flows. We can develop a structural view of a process or system using this method. Before beginning a project or system, all developers must design a business process model.

A user can create a new task from my system's task creation page after confirming the completion on any given date. The administrator can then cancel or confirm the new task. The business process model for this process is shown in Figure 3.1.

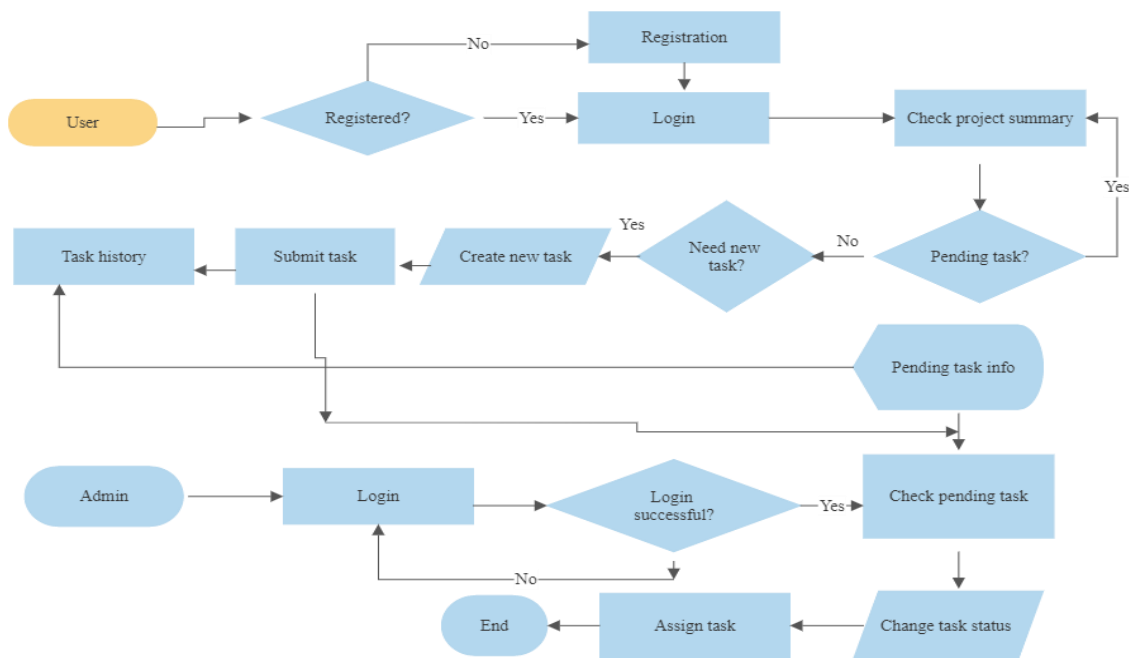


Figure 3.1: Task Process Business Process Model

### **3.2 Requirement Collection & Analysis**

For any developing project, we need to collect requirements for the initial work. I need to make a graphical representation of the project. Without a graphical view, it is difficult to implement a project. I've had a lot of trouble gathering criteria. For me, it was a huge undertaking. Because I was collecting requirements by myself, it was even more difficult when I realized that criteria were not available everywhere. As a result, there's a potential you'll get incorrect information. As a result, I'll need to gather data from reliable sources. In addition, I must be extremely cautious when gathering requirements. Customer project and task management system is the name of my project. I gathered all of the requirements before beginning to construct my project. I gathered data from a variety of sources. I looked for task management websites and websites that provide project management knowledge in Bangladesh. I need to gather information and make a personal visit to a software company. I need to gather the contact information for managers and call them to gain more precise and accurate information about customers and owner needs. I was familiar with a software company's work procedure and offerings. I analyzed the requirements I gathered. Following my study, I created a number of method models to represent the system. To complete a successful project, we must first gather requirements, then assess those requirements and choose the project's technique.

### **3.3 Use Case Modeling and Description**

A behavioral UML diagram is a use case diagram. A use case diagram is a tool that may be used to analyze any system. These illustrations demonstrate how to identify different sorts of roles in a system and interact with them, allowing us to better understand how beneficial they are. At one point, the diagram's purpose assists us in identifying the functions and how to perform those duties. Any project's development requires the use of use case diagrams. A high-level representation of the system is a use case diagram. This diagram helps to identify both internal and external causes [7].

We can see this in Figure 3.2 For users, use case studies.

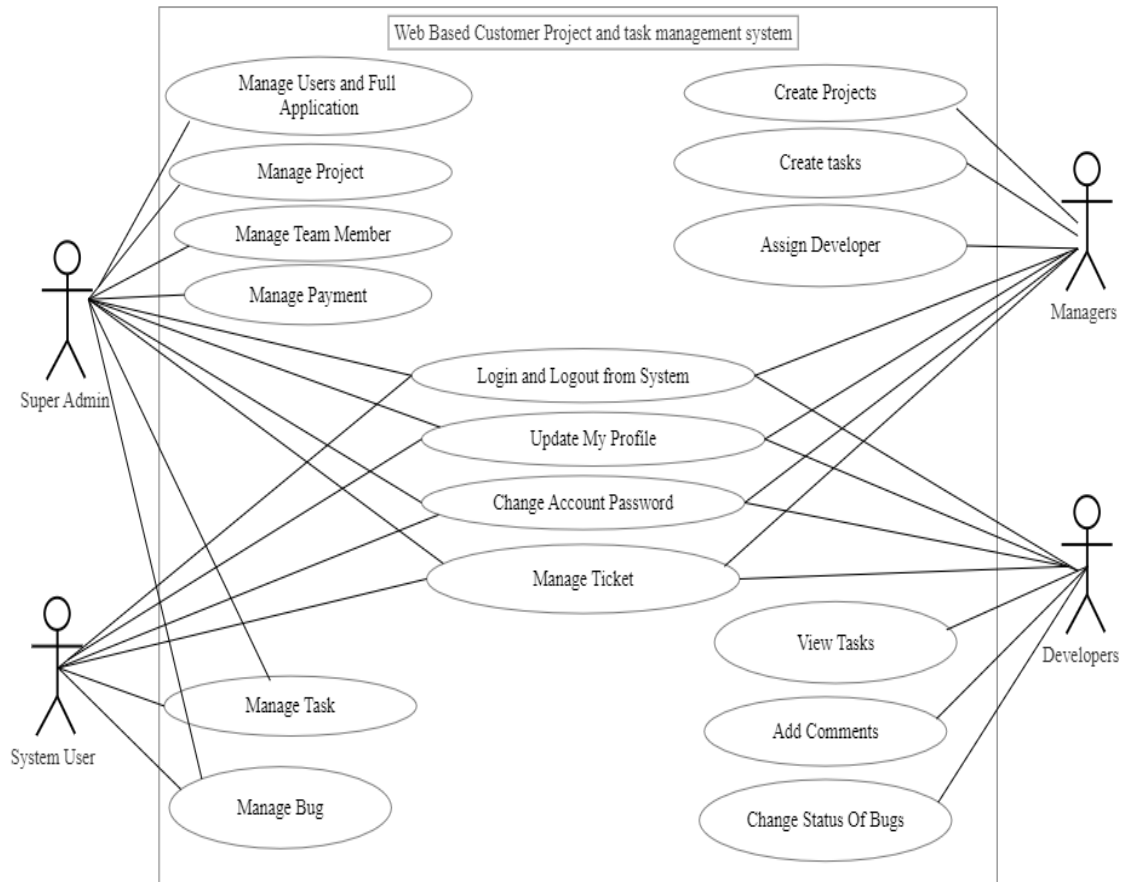


Figure 3.2: Users Use Case Model

Figure 3.2 depicts the several user functions implemented in my project. Signing in is an option for a system user. They can also sign into the system to log in. They can check their profile and change the information they supplied in their registration form after logging in. They can manage their tasks and bugs from the task section. They can also submit a ticket or respond to admin or legitimate authority. Admin has access to the support ticket. They also have the option of changing their password.

A manager can also sign in, as seen in Figure 3.2. They can check their profile and change the information they supplied in their registration form after logging in. They can establish

projects and project tasks from the project section, as well as assign developers to project tasks. A developer can also sign in, as seen in Figure 3.2. They can check their profile and change the information they supplied in their registration form after logging in. They can examine and comment on tasks from the task section. They can also change the status of a task. In this diagram, figure 3.2, I'm attempting to demonstrate that there are numerous administrative services implemented in my project. The admin's activities are depicted in this figure. Admins can manage team members, projects, tasks, and payments, among other things. The use case model in figure 3.2 shows that Admin has a plethora of other possibilities.

### **3.4 Logical Data Model**

We require reporting in the business world, and we may report database items using the Logical Data Model. The basic elements of ER models are entities and their common relationships [8].

A logical data model is the Entity Relationship Diagram (ERD). Figure 3.3 depicts the ER diagram for my project.

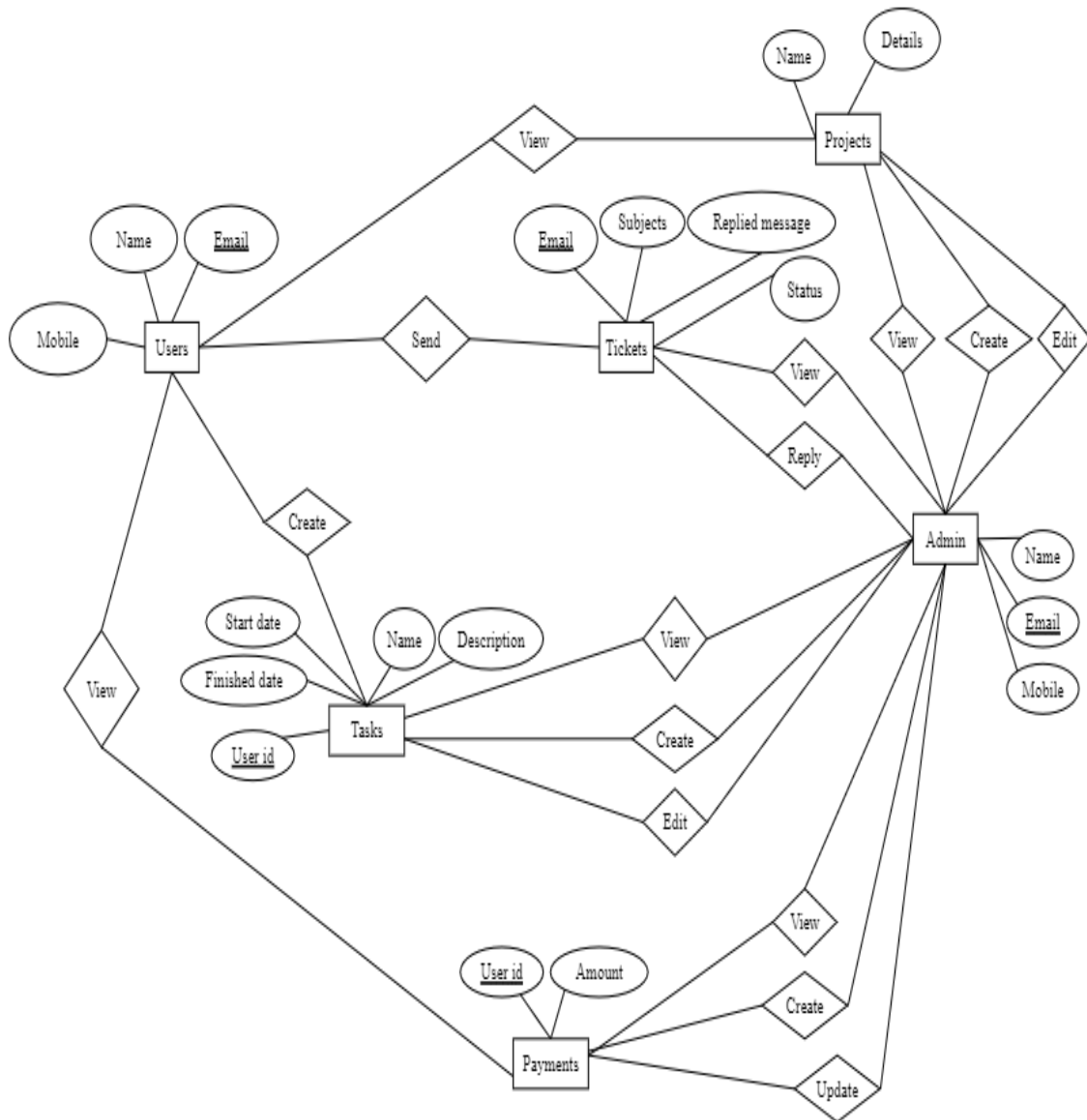


Figure 3.3: The system's ER diagram

### 3.5 Design Requirements

- Customers and administrators are the two types of users in this system.
- Users can register or sign up as a general user.
- The panel also allows admin users to sign up.
- Users can display their profile information after logging in.
- Customers can get updates on the project's progress.
- A client user can monitor the project, assign task status, and create new tasks after logging in.
- A customer user can check payment status and also pay the due amount to the admin.
- A client user will be able to submit an admin a help ticket. In addition, the administrator responds to the communication.
- Customers and team members can be created by an authentic administrator.
- An admin is able to create, edit & delete customer projects.
- An admin can create, edit & delete project tasks for customers.
- An admin can also assign tasks to the team members.
- The database allows the administrator to view pending and finished task lists.
- Admin can check project payment conditions and also create, edit invoices.
- Admin is able to check support tickets and also can reply to the message to a customer.
- Admin can check an assign member timesheet.

I tried to keep the project simple and user-friendly for all users when I first started designing it. It's vital to be efficient. As a result, I keep it in mind in order for my project to work properly and smoothly on the web sites.



## CHAPTER 4

### DESIGN SPECIFICATION

#### 4.1 Front-End Design

The front-end design of any application is critical since it is how people interact with it. We made every effort to make it user-friendly and clear.

We utilized HTML (Hypertext Markup Language) to design the user interface. We needed to create each and every page of our application because it has so many. Let's have a look at a few pages to see how they were designed and coded.

```
@endif
@endforeach
<div class="form-group has-feedback" id="name-group">
  <input type="text" class="form-control" placeholder="First Name" name="first_name" required='required' value="" />
  @if($errors->has('first_name'))
    <p class="text-danger">{{ $errors->first('first_name')}}</p>
  @endif
</div>
<div class="form-group has-feedback" id="name-group">
  <input type="text" class="form-control" placeholder="Last Name" name="last_name" required='required' value="" />
  @if($errors->has('last_name'))
    <p class="text-danger">{{ $errors->first('last_name')}}</p>
  @endif
</div>
<div class="form-group has-feedback" id="email-group">
  <input type="email" class="form-control" placeholder="Email" name="email" required='required' value="" />
  @if($errors->has('email'))
    <p class="text-danger">{{ $errors->first('email')}}</p>
  @endif
</div>
<div class="form-group has-feedback" id="password-group">
  <input type="password" class="form-control" placeholder="Password" name="password" required='required' />
  @if($errors->has('password'))
    <p class="text-danger">{{ $errors->first('password')}}</p>
  @endif
</div>
```

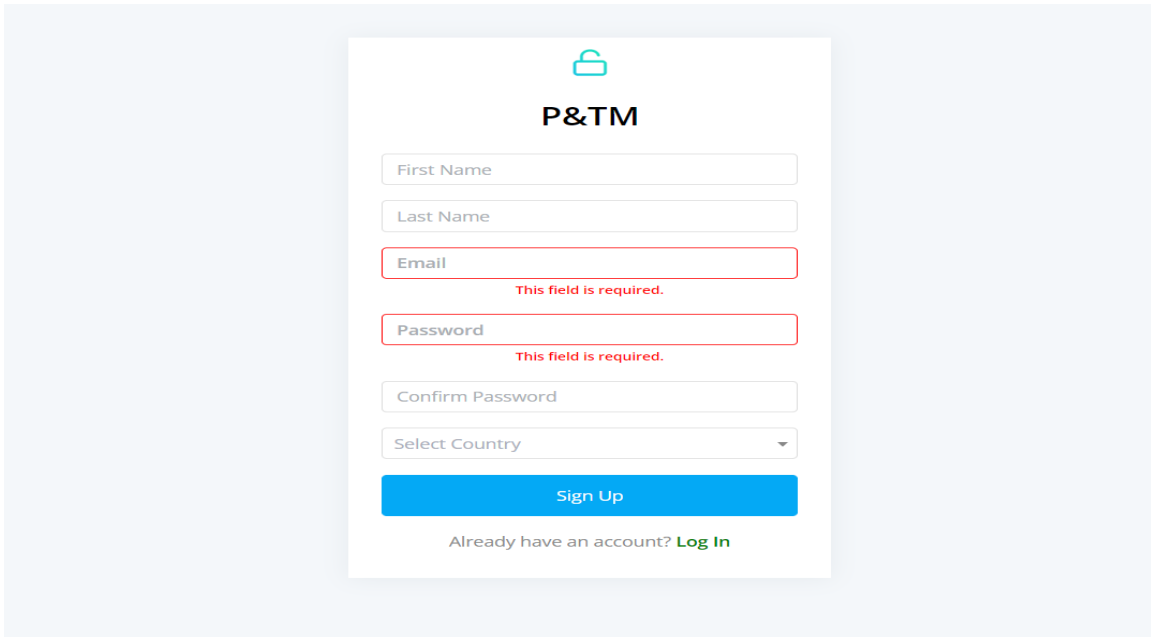


Figure 4.1: Code and Design for the Customer Registration Page.

The client registration page, shown in Figure 4.1, is a good example of this. HTML and CSS are used to create it.

However, because I want this report to be as smart as possible, I did not supply the complete html and CSS code. I also use jquery to accomplish the actions on this website.

```

<h5 class="text-white text-center">{{ __('Logged Times') }}</h5>
</div>
</div>
</div>
<div class="col-md-6">
<div class="progress">
<div class="circle" data-percent="{{ !empty($totalTask) ? (100 * $completedTask) / $totalTask : 100 }}">
<div></div>
<p class="text-7">{{ __('Project Progress') }}</p>
</div>
</div>
</div>
<div class="media">
<div class="photo-table">
<h5 class="mb-2 f-16 f-w-200 reslate">{{ timeZone.getTime($project->begin_date) }}</h5>
<h6 class="text-muted mb-0">{{ __('Start Date') }}</h6>
</div>
<div class="media-body">
<div class="float-right">
<h5 class="mb-2 f-16 f-w-200 reslate">{{ isset($project->due_date) ? timeZone.formatDate($project->due_date) : '' }}</h5>
<h6 class="text-muted mb-0">{{ __('End Date') }}</h6>
</div>
</div>
</div>
<div class="media b-top-ddd">
<div class="photo-table">
<h6>{{ __('Charge Type') }}</h6>
</div>
<div class="media-body">
<span class="f-16 f-w-400 ml-2 float-right color-black">
@if($project->charge_type == 1)
{{ __('Fixed Rate') }}
@elseif($project->charge_type == 2)
{{ __('Project Hours') }}
@elseif($project->charge_type == 3)
{{ __('Task Hours') }}
@endif
</span>
</div>
</div>
</div>

```

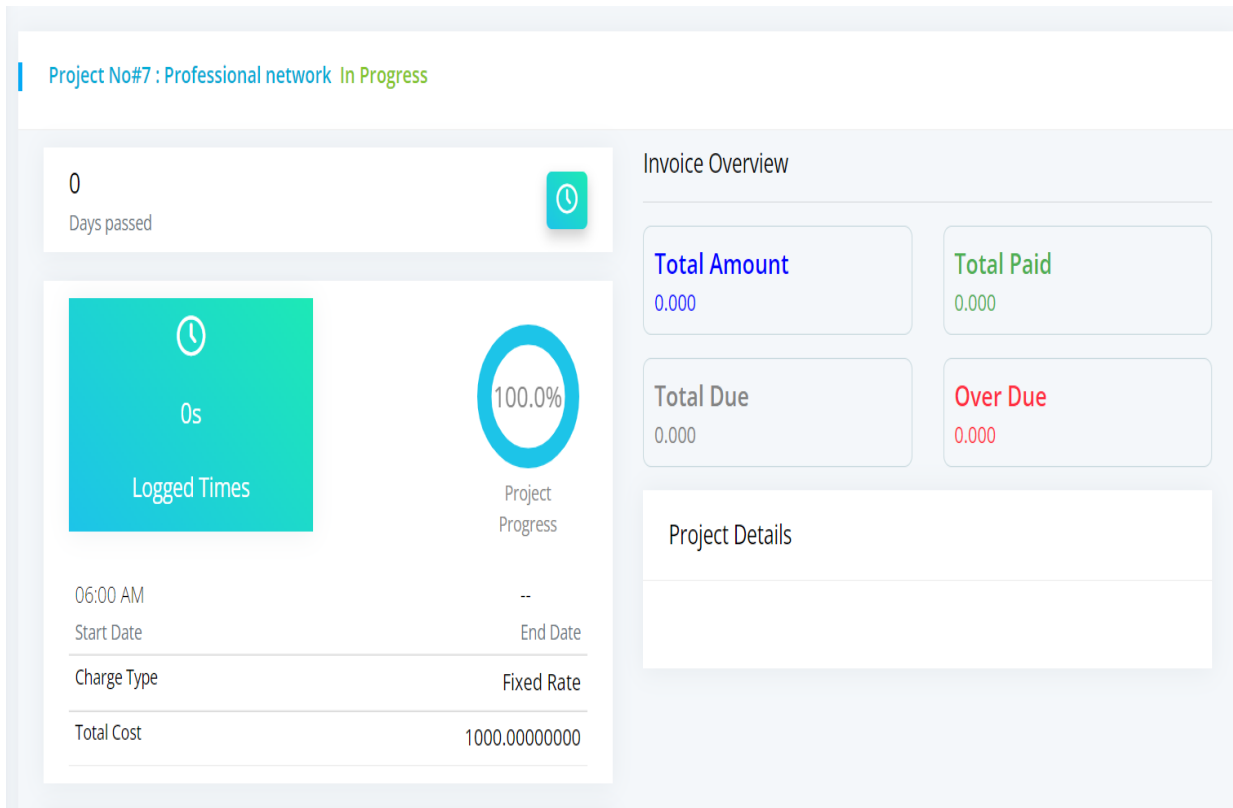


Figure 4.2: Customer Project Detail Page for Customer

The design of our customer project detail page is as follows. In Figure 4.2, I've only shown a portion of the HTML and CSS code.

```

<form action="">
<div class="wrapper admin_home">
<div class="one">
<h1><i>Admin Home Pages</i></h1>
</div>
<div class="two">

<div class="package">
<a href="#"><i class="fa fa-cullery" aria-hidden="true"></i></a>
<a href="#" style="padding-left: 5px;">Check Package</a>

</div>
<div class="check">
<a href="#"><i class="fa fa-calendar-check-o" style="color: #1543de"></i></a>
<a href="#" style="padding-left: 35px;">Check Booking</a>

</div>
<div class="pending">
<a href="#"><i class="fa fa-hourglass-half"></i></a>
<a href="#" style="font-size: 15px; padding-left: 5px;">Pending Booking Request</a>

</div>
</div>
<div class="three">

<div class="package">
<a href="#"><i class="fa fa-check-square" style="color: green;"></i></a>
<a href="#" style="font-size: 15px; padding-left: 5px;">Confirmed Booking Request</a>

</div>
<div class="check">
<a href="#"><i class="fa fa-picture-o" style="color: aqua;"></i></a>
<a href="#" style="padding-left: 35px;">Upload Picture</a>

</div>
<div class="pending">
<a href="#"><i class="fa fa-trash" style="color: #1543de; padding-left: 18px;"></i></a></div>

<div class="check" style="margin-left: 350px;">
<a href="#"><i class="fa fa-sign-in" style="color: #18a8e8;"></i></a>
<a href="#"><input type="submit" value="Log in" style="width: 80px; font-size: 20px; font-family: sans-serif; color: blue; font-weight:600; border-radius: 5px;border: 0px; margin-top: 0px; margin-left: 10px; cursor: pointer;text-transform: capitalize;"></a>
</div>

e31818
-->
</div>
<div class="six">

<a href="https://www.facebook.com/log-in" target="_blank"> <i class="fa fa-facebook" style="background: #fff; padding-left: 20px; color: #1543de; width: 50px; height: 50px; margin-left:70px; font-size: 20px; line-height: 60px; border-radius: 50%; margin-top: 50px;"></i></a>

<a href="https://www.twitter.com/log-in" target="_blank"> <i class="fa fa-twitter" style="background: #18c7e6; padding-left: 15px; color: #fff; width: 50px; height: 50px; margin-left:20px; font-size: 20px; line-height: 60px; border-radius: 50%;"></i></a>

<a href="https://www.twitter.com/log-in" target="_blank"> <i class="fa fa-instagram" style="background: #e36618; padding-left: 15px; color: #fff; width: 50px; height: 50px; margin-left:20px; font-size: 20px; line-height: 60px; border-radius: 50%;"></i></a>

<a href="https://www.google.com/" target="_blank"> <i class="fa fa-google-plus" style="background: #e31818; padding-left: 15px;color: #fff; width: 50px; height: 50px; margin-left:20px; font-size: 20px; line-height: 60px; border-radius: 50%;"></i></a>
</div>

```

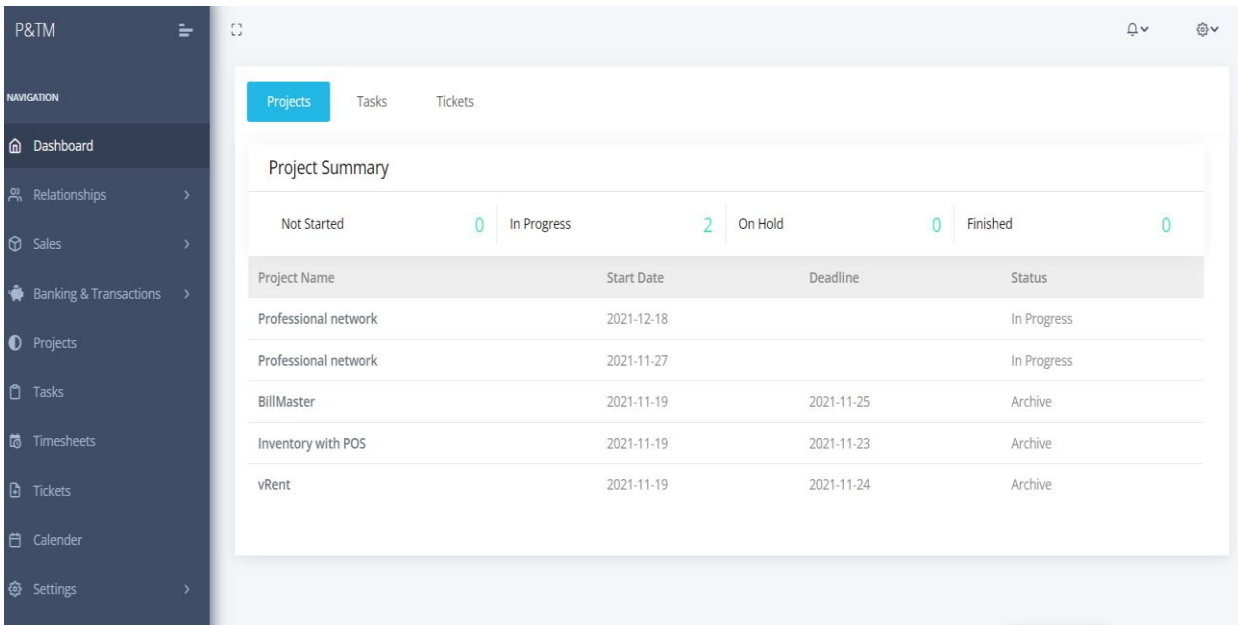


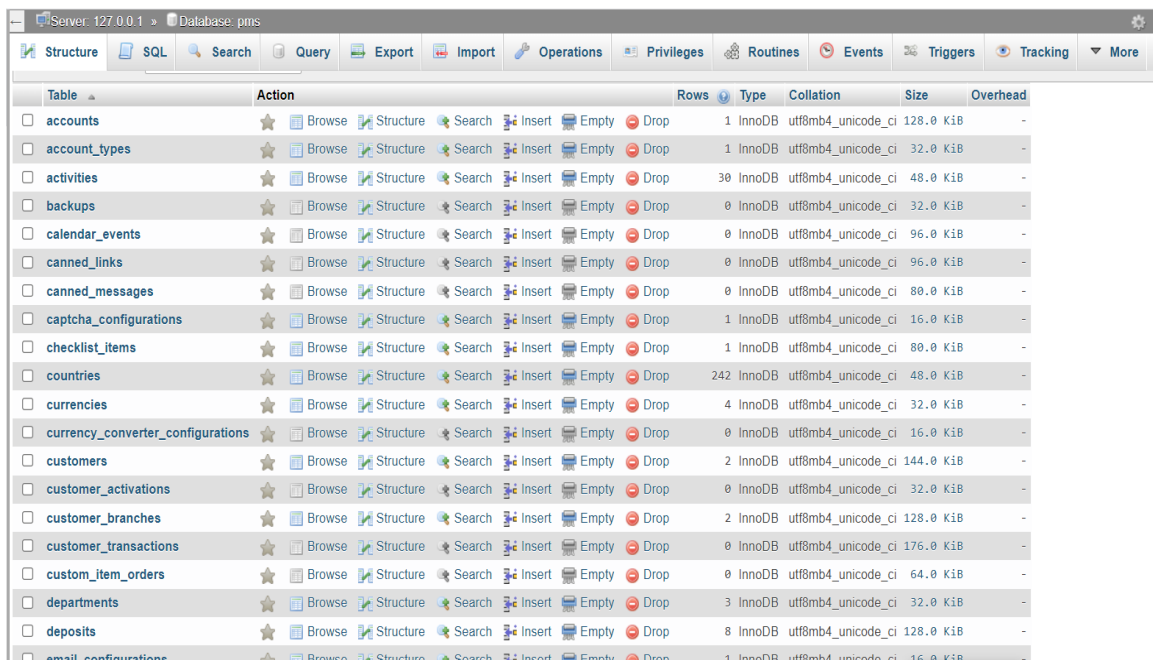
Figure 4.3: Web Browser Admin Dashboard Page

Figure 4.3 depicts the layout of this project's admin dashboard page, as well as some HTML and CSS code. Because this project has so many pages, each page has its own HTML and CSS code. However, it's vital to remember that these sites won't work without PHP code, which I haven't included here.

## 4.2 Back-End Design

This project's Back-End Part has completed all of the project's logical tasks. Database connection, database use, database maintenance, and so on are all handled in the back-end section. For the back-end coding of my project, I utilized PHP (Laravel framework) as the programming language.

I used a MySQL database for this project's back-end maintenance. A user authentication system is provided by MySQL.



The screenshot shows a MySQL database management interface. The top bar indicates the server is '127.0.0.1' and the database is 'pms'. The interface includes a menu bar with options like Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, Tracking, and More. Below the menu is a table listing database tables with columns for Table, Action, Rows, Type, Collation, Size, and Overhead. The table lists various tables such as accounts, account\_types, activities, backups, calendar\_events, canned\_links, canned\_messages, captcha\_configurations, checklist\_items, countries, currencies, currency\_converter\_configurations, customers, customer\_activations, customer\_branches, customer\_transactions, custom\_item\_orders, departments, deposits, and email\_configurations.

Table	Action	Rows	Type	Collation	Size	Overhead
accounts	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	128.0 KiB	-
account_types	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
activities	★ Browse Structure Search Insert Empty Drop	30	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
backups	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
calendar_events	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	96.0 KiB	-
canned_links	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	96.0 KiB	-
canned_messages	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
captcha_configurations	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
checklist_items	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
countries	★ Browse Structure Search Insert Empty Drop	242	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
currencies	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
currency_converter_configurations	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
customers	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	144.0 KiB	-
customer_activations	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
customer_branches	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	128.0 KiB	-
customer_transactions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	176.0 KiB	-
custom_item_orders	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-
departments	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
deposits	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_unicode_ci	128.0 KiB	-
email_configurations	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-

Figure 4.4: The Home Page for the Database

The database's primary page is shown in Figure 4.4. Here's a database table. Customers, countries, departments, projects, tasks, task assigns, tickets, and users are just a few examples. For this database's authentication page, an administrator can add new features and a new table. In addition, an administrator has the ability to remove tables or features from this area.

	id	migration	batch
<input type="checkbox"/>	1	2020_04_06_093559_create_account_types_table	1
<input type="checkbox"/>	2	2020_04_06_093559_create_accounts_table	1
<input type="checkbox"/>	3	2020_04_06_093559_create_activities_table	1
<input type="checkbox"/>	4	2020_04_06_093559_create_backups_table	1
<input type="checkbox"/>	5	2020_04_06_093559_create_calendar_events_table	1
<input type="checkbox"/>	6	2020_04_06_093559_create_checklist_items_table	1
<input type="checkbox"/>	7	2020_04_06_093559_create_countries_table	1
<input type="checkbox"/>	8	2020_04_06_093559_create_currencies_table	1
<input type="checkbox"/>	9	2020_04_06_093559_create_custom_item_orders_table	1
<input type="checkbox"/>	10	2020_04_06_093559_create_customer_activations_tabl...	1
<input type="checkbox"/>	11	2020_04_06_093559_create_customer_branches_table	1
<input type="checkbox"/>	12	2020_04_06_093559_create_customer_transactions_tab...	1
<input type="checkbox"/>	13	2020_04_06_093559_create_customers_table	1
<input type="checkbox"/>	14	2020_04_06_093559_create_departments_table	1
<input type="checkbox"/>	15	2020_04_06_093559_create_deposits_table	1
<input type="checkbox"/>	16	2020_04_06_093559_create_email_configurations_tabl...	1
<input type="checkbox"/>	17	2020_04_06_093559_create_email_templates_table	1
<input type="checkbox"/>	18	2020_04_06_093559_create_exchange_rates_table	1
<input type="checkbox"/>	19	2020_04_06_093559_create_expenses_table	1
<input type="checkbox"/>	20	2020_04_06_093559_create_files_table	1

Figure 4.5: Page of Migration

The migration database page is shown in Figure 4.5. This database page is linked to PHP (Laravel Framework). PHP(Laravel FrameWork) is used to generate and connect each table database.

	id	full_name	first_name	last_name	added_by	password	role_id	phone	email	remember
<input type="checkbox"/>	1	Ashrafuzzaman Ashraf	Ashrafuzzaman	Ashraf	1	\$2y\$10\$0rqMqCDvT6AFi0h1fJHz9OCKEjohBffjAWWK3FVA6m9...	1	NULL	ashraf@info.com	
<input type="checkbox"/>	6	Juwel Isalm	Juwel	Isalm	1	\$2y\$10\$Hp8lmokYaZ0J6QrzqTUQS.pS9m/IRp3IEHYVg5INmF...	1	NULL	juwel@gmail.com	

Figure 4.6: User Page Archiving

Every user in our database is in charge of account storage is shown in Figure 4.6. This database stores their profile, photo, secure password, and address.

## 4.3 Interaction Design and UX

### Design for the User's Home Page:

- Project Info (Information on the project's tasks can be found here)
- Payment Info. (Payment information can be found here)
- Support ticket (Users have the ability to send tickets. The administrator's response is displayed below, and users can remove his own message).
- Make a new account (Create an account by registering on the website)
- Sign Up and Sign In
- The Bar with Menus (Dashboard, Quotations, Invoices, Payments, Projects, Supports)

### Design of the Admin Home Page:

- Sign In (The sign in form allows admins to log in)
- Home Page for Admins (Project, Task, and Ticket Lists in Preview)
- Manage Customers (Customers can be added, edited, and deleted by the administrator)
- Manage Team Members (Team Members can be added, edited, and deleted by the administrator)
- Manage Project (Admins can add, edit, delete, and assign members to projects)
- Manage Task (Admin can add, modify, delete, and assign members to tasks)
- Pending Task Request (Cancel, Details are displayed in Progress; sort by completion date and generated date)
- Manage Payment (Payment list, add, modify, and delete are all options available to the administrator)
- Upload File (Admins have the ability to upload files to profiles, projects, and tasks)
- Delete File (Admins have the ability to delete files from profiles, projects, and tasks)
- Respond to the message on the ticket (Admins can view and respond to user ticket messages, as well as delete them)
- Timesheet Management (Add, Edit, and Delete)

- Manage Setting (Lists can be created, edited, and deleted by the administrator)
- Manage Calendar (Admins have the ability to add events to the calendar)
- Please log out (logging out of the sign-out form)

## **UX**

The abbreviation UX stands for User Experience. I attempted to create a user-friendly website. This website has been accessed from a number of different browsers. This project has been utilized by a few people, and they have all given me great feedback. Everything seems to be in order. However, because I haven't yet made this website public, I don't have enough data on user experience at this time.

## **4.4 Implementation Requirements**

Any PHP-related development project can be completed using a variety of frameworks or languages. Because it looked more convenient to us, I utilized PhpStorm and Visual Studio. PHP(Laravel Framework) has a large number of built-in classes. As a result, it simplifies your coding. HTML, CSS, and the JAVASCRIPT Library JQuery make design a breeze. The next sections explain the requirements for implementing our project:

- This project relies heavily on the PHP Laravel Framework. It is a complete kit for creating a website application. I utilized the most recent version of the PHP Laravel Framework, which is version 8.12.0. Despite the fact that I began my project with an older version, I later upgraded it to version 8.12.0.
- Servers aren't a concern for us. MySQL provides us with the ability to use servers. MySQL's storage, database, and authentication functions are all very easy to use. MySQL is in charge of the server, hosting, and so on. My project requires a MySQL connection.
- Emulator is a PhpStorm tool that allows you to create a virtual operating system in order to debug, test, and execute your application on different browsers.



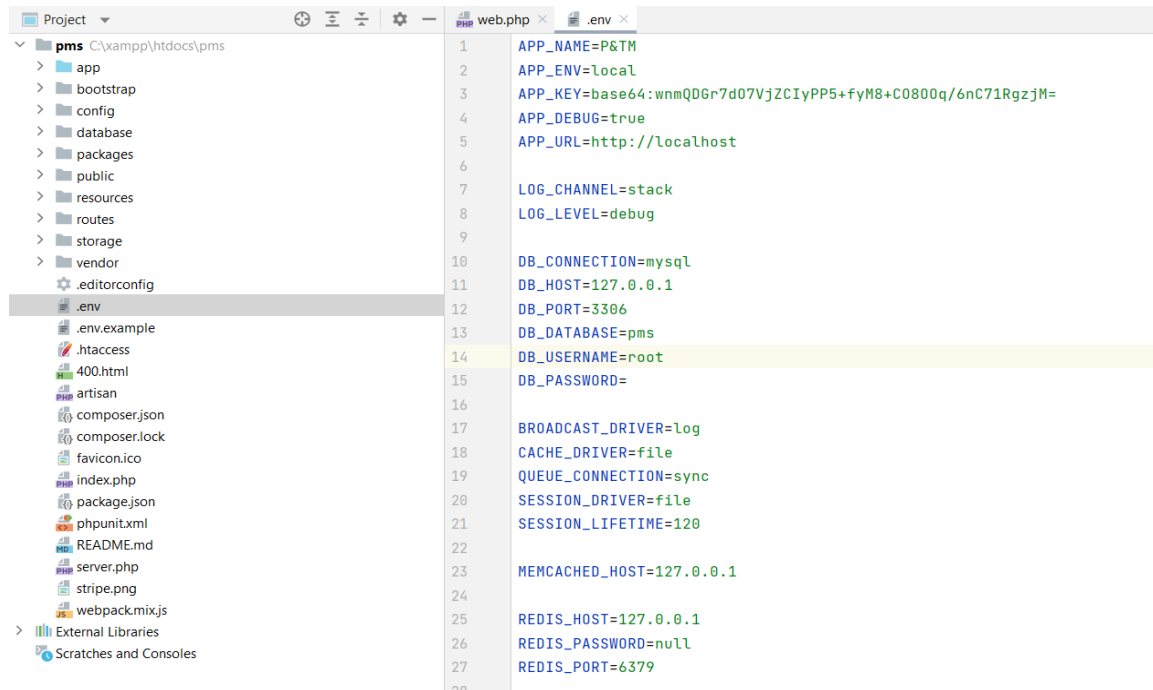
## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation of Database

For any management system project, a database is a must-have. In our database, we need to store a variety of information. We save data in a database and retrieve it as needed. Despite the fact that I was working on a web application, I needed a database. MySQL is a database management system. I don't need to worry about the server or hosting if I utilize a MySQL database. All of these are handled by MySQL.

First and foremost, I connected my project to MySQL by following the instructions on the MySQL website. Different databases must be set up in MySQL. After that, we must connect it to PHP code. After connecting to the php server and confirming authentication. Set up the database name in the ENV file, migrate it in Laravel, and import all the tables from the database.



```
1 APP_NAME=P&TM
2 APP_ENV=local
3 APP_KEY=base64:wnmQDGr7d07VjZCIyPP5+fyM8+C0800q/6nC71RgzjM=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=pms
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 QUEUE_CONNECTION=sync
20 SESSION_DRIVER=file
21 SESSION_LIFETIME=120
22
23 MEMCACHED_HOST=127.0.0.1
24
25 REDIS_HOST=127.0.0.1
26 REDIS_PASSWORD=null
27 REDIS_PORT=6379
```

Figure 5.1: PHP (Laravel Framework) Code for Database Connection

The PHP (Laravel) code has linked to the database, as seen in Figure 5.1. It is used to authenticate MySQL and create storage. It facilitates the creation of a table or several database tables in MySQL and their connection with PHP code.

The screenshot shows a MySQL database management interface with the following table list:

Table	Action	Rows	Type	Collation	Size	Overhead
accounts	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	128.0 KiB	-
account_types	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
activities	★ Browse Structure Search Insert Empty Drop	30	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
backups	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
calendar_events	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	96.0 KiB	-
canned_links	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	96.0 KiB	-
canned_messages	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
captcha_configurations	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
checkboxlist_items	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	80.0 KiB	-
countries	★ Browse Structure Search Insert Empty Drop	242	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
currencies	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
currency_converter_configurations	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
customers	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	144.0 KiB	-
customer_activations	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
customer_branches	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	128.0 KiB	-
customer_transactions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	176.0 KiB	-
custom_item_orders	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-
departments	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
deposits	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_unicode_ci	128.0 KiB	-
email_configurations	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-

Figure 5.2: Database: MySQL

The MySQL Database page is shown in Figure 5.2. The database table can be seen here. Customers, team members, projects, tasks, task assignments, tickets, ticket replies, and users are just a few examples. For this database's authentication page, an administrator can add new features and a new table. In addition, an administrator has the ability to remove tables or features from this area.

## 5.2 Implementation of Front-End Design

Making a web application's user interface basic and easy to comprehend is critical to its success. If it is not basic and understandable, people will not be interested. I keep this procedure in mind and create an appealing user interface for the clients. In this project, there are two types of modules. There are two types of modules: user and admin.

People will not want to utilize this if it is difficult to grasp. We made every effort to create a user-friendly interface. In our project, we have two different sorts of modules: user and admin. Figure shows the user and admin modules' user interfaces.

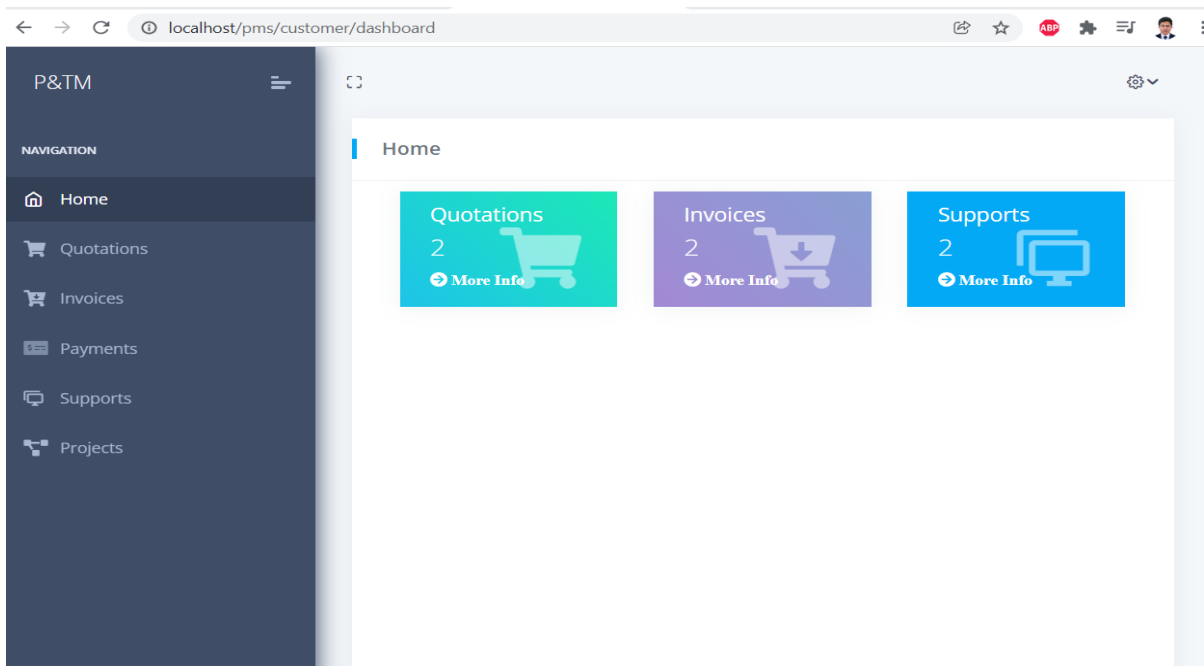


Figure 5.3: Home Page of the User

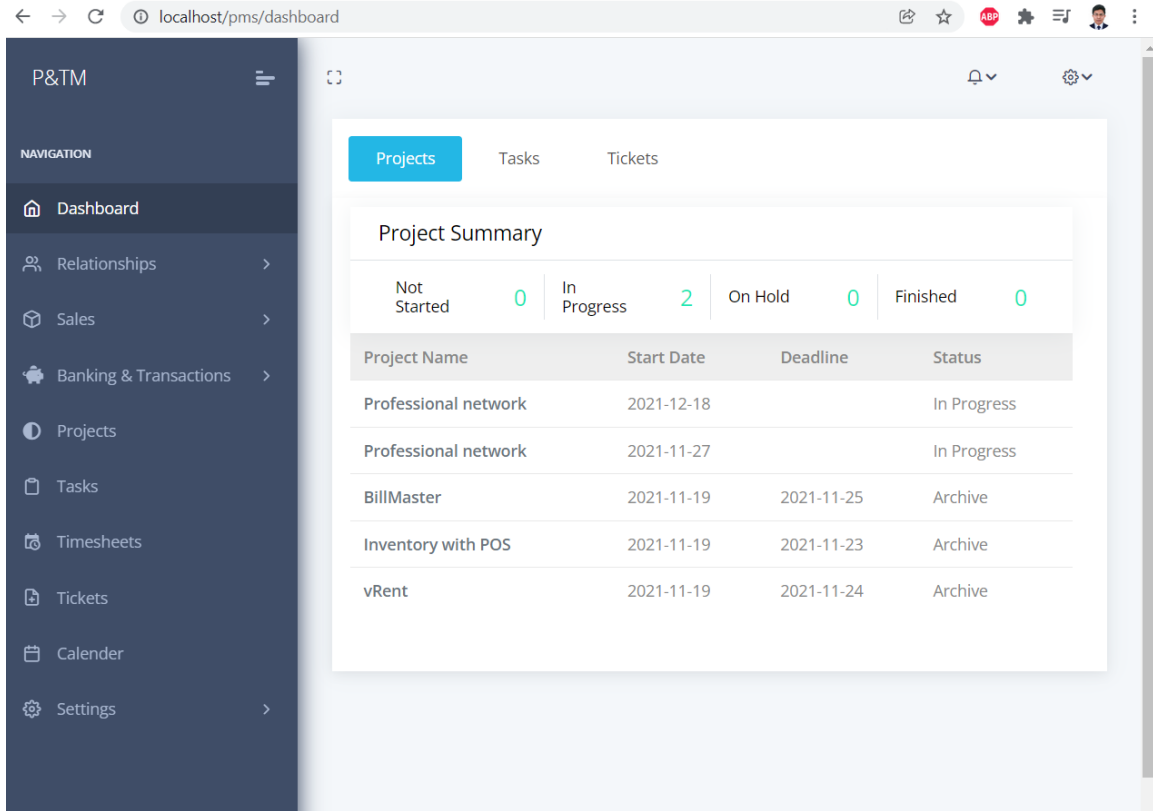


Figure 5.4: Home Page for Admins

Two alternate project home pages are shown in Figures 5.3 and 5.4. Many features for the users have been built in these two modules. I'll start with the customer's user features and then move on to the admin features.

### 5.2.1 User Module

Figure 5.3 does not represent all of the functionality available in the User Module. As a result, I've included a list of the module's other features below. User login page shown in Figure 5.5. User registration page shown in Figure 5.6. In Figure 5.7 shown in home page after a user login. In Figure 5.8 shown in user profile update page.

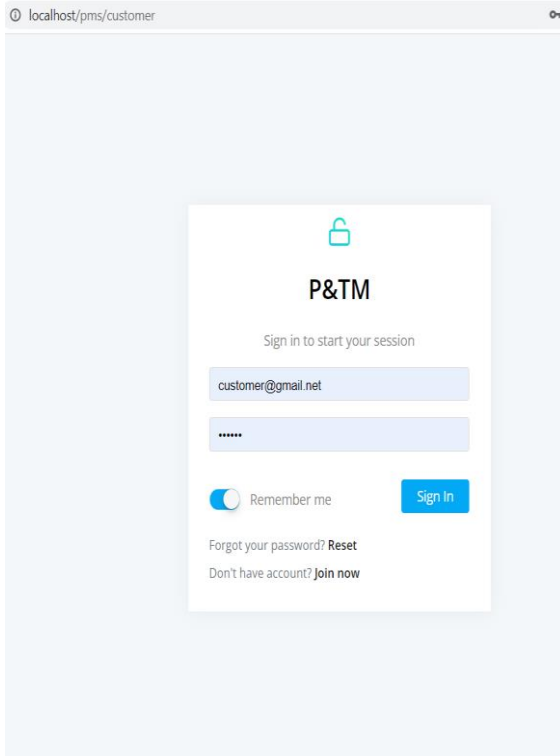


Figure 5.5: Login Page for Users

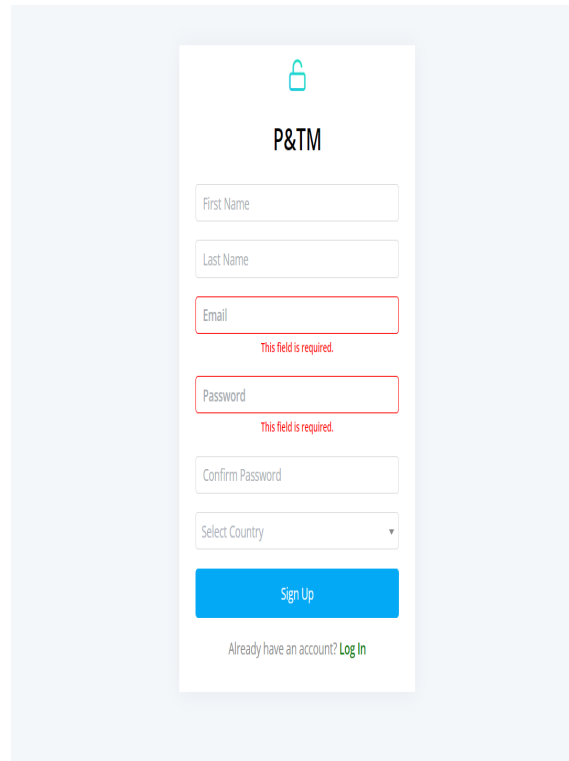


Figure 5.6: Page for User Registration

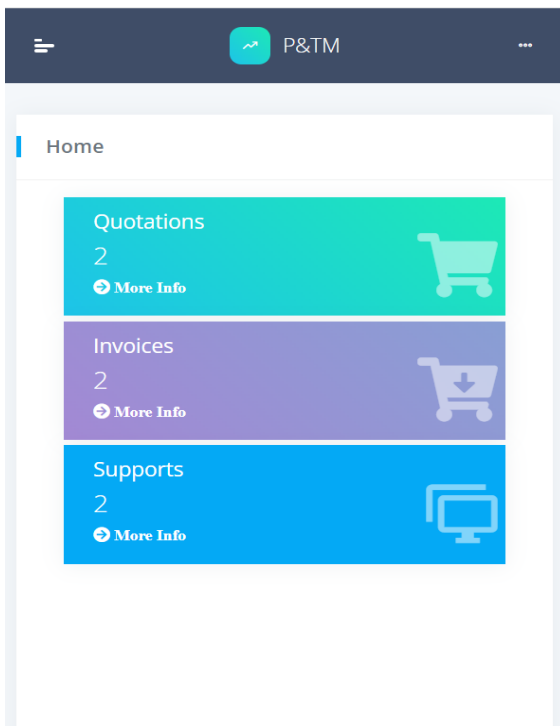


Figure 5.7: After you have logged in

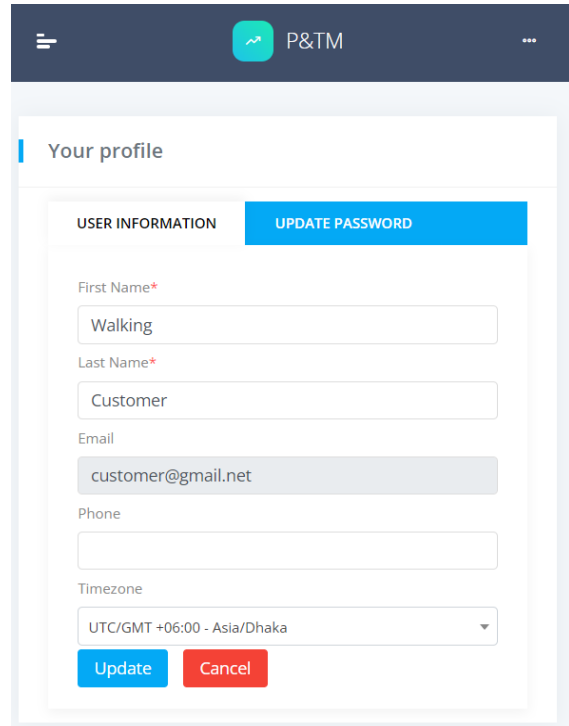


Figure 5.8: Updates to User Profiles

A user will be able to register on this website after providing all of the necessary information.

A user will be able to log in to this system after entering a valid email address and password. If a user provides the incorrect email and password, he or she will be unable to log in to the website. In addition, the window will display an alert warning message. Customers will be able to change their passwords and update their user profiles if they forget theirs. Figure 5.8 illustrates this.

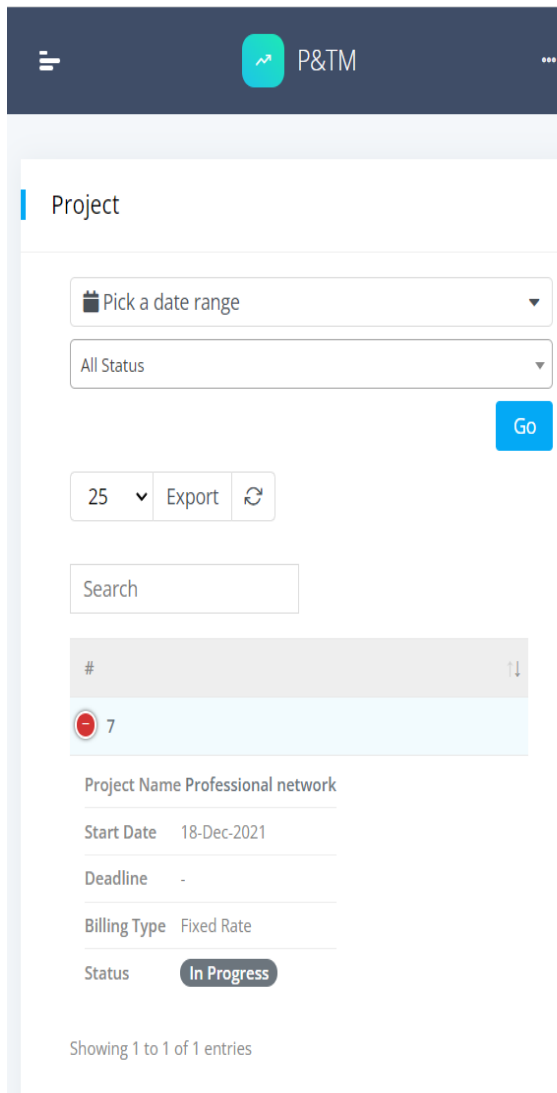


Figure 5.9: Customer Project

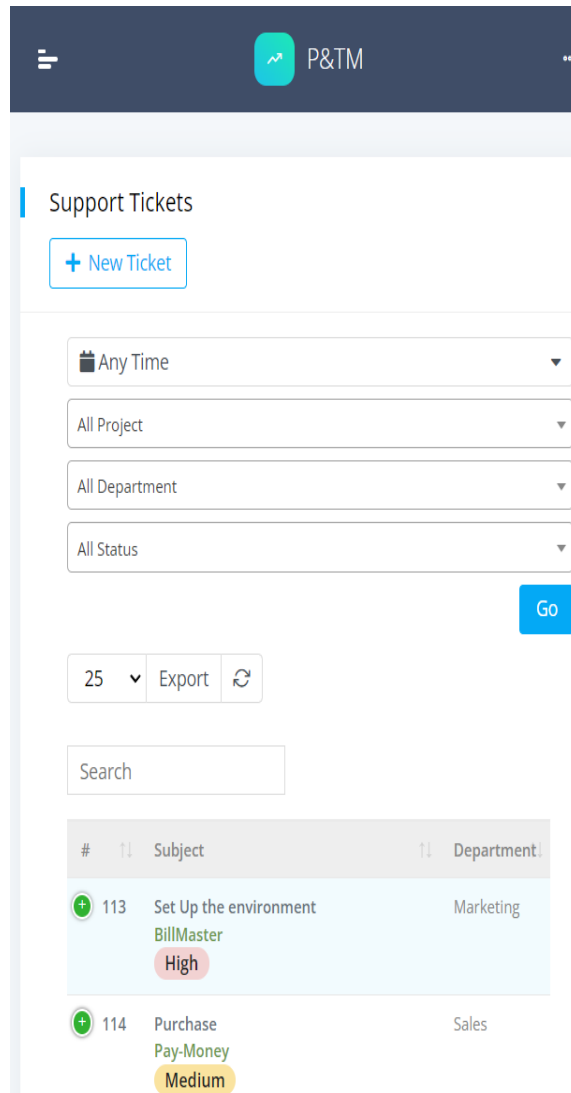


Figure 5.10: Customer Support Ticket

Figure 5.9 depicts the application's customer project list, which includes projects generated by the website's administrator.

We exhibited example support tickets created by customers in Figure 5.10. I utilized some tickets here, however clients can create more tickets if they need them on this website.

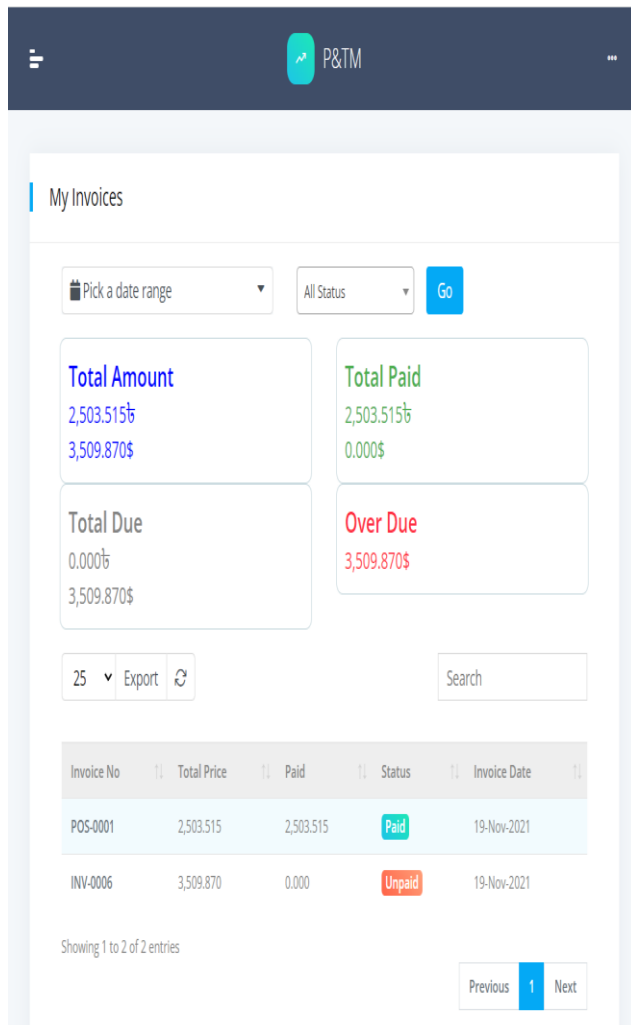


Figure 5.11: Customer Invoices

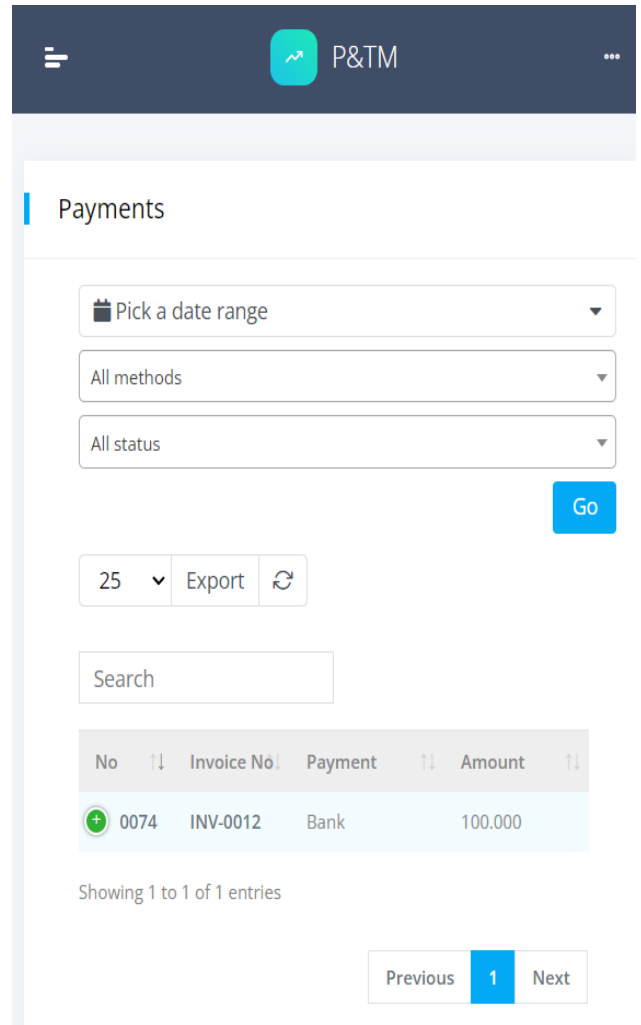


Figure 5.12: Customer Payment

Figure 5.11 showed us all the customer paid and unpaid invoices list and the customer can view invoice details. And other Figure 5.12 showed us payment list pages for customers.

The screenshot shows the 'Open Ticket' form in the P&TM system. The header includes a menu icon and the P&TM logo. The main heading is 'Open Ticket'. Below it, there is a 'TICKET INFORMATION' section with a blue highlight. The form contains several fields: 'Subject\*' (text input), 'Project' (dropdown menu with 'Select One' selected), 'Message\*' (rich text editor with a toolbar containing Paragraph, Bold, Italic, Link, Unlink, Bulleted List, Numbered List, Image, Quote, and Undo), and 'Department\*' (text input). A 'Canned Message' button is also visible.

Figure 5.13: Customer New Ticket

The screenshot shows the 'Customer Ticket Reply' form in the P&TM system. The header includes a menu icon and the P&TM logo. The main heading is 'Ticket No: #113' with an 'Open' status indicator. Below it, the 'Subject' is 'Set Up the environment'. There are three tags: 'Department: Marketing', 'Priority: High', and 'Assignee: Ashrafuzzaman Ashraf'. The form includes a 'Canned Message' button, a 'Canned Links' button, and a 'Reply\*' section with a rich text editor (same toolbar as Figure 5.13) and a 'File' section with a '+' icon for uploading.

Figure 5.14: Customer Ticket Reply



A form page for creating a support ticket was shown in Figure 5.13. If a user requires assistance with a project, they can create a ticket. Figure 5.14, on the other hand, depicts the Users' reply ticket list and form.

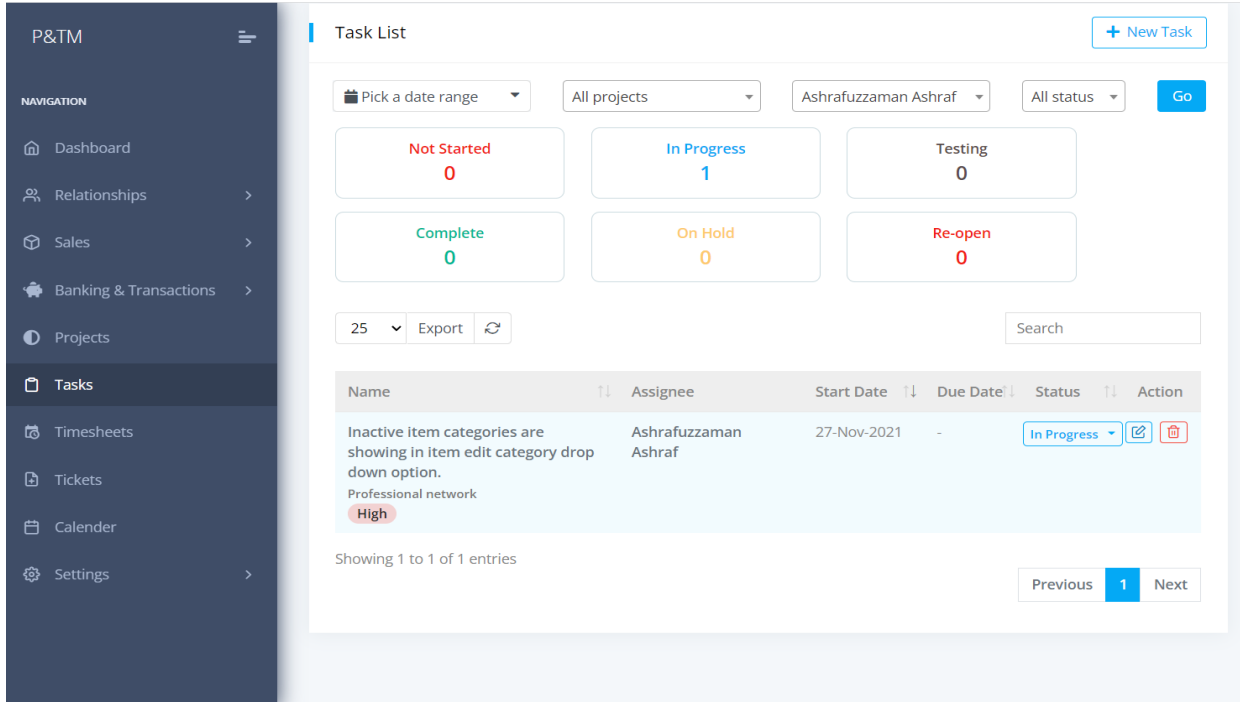


Figure 5.15: Admin Task List.

The task list by user is shown in Figure 5.15. This set of client tasks could be accepted by an admin. The DELETE action allows the admin to remove tasks from the list. A list of clients who are creating project tasks has been generated at random. Customer data is gathered in its entirety. This table can be viewed in the Database by the Admin.

## 5.2.2 Admin Modules

Figure 5.4 depicts the features of the Admin Modules. More information is available here. Below this section is a list of resources:

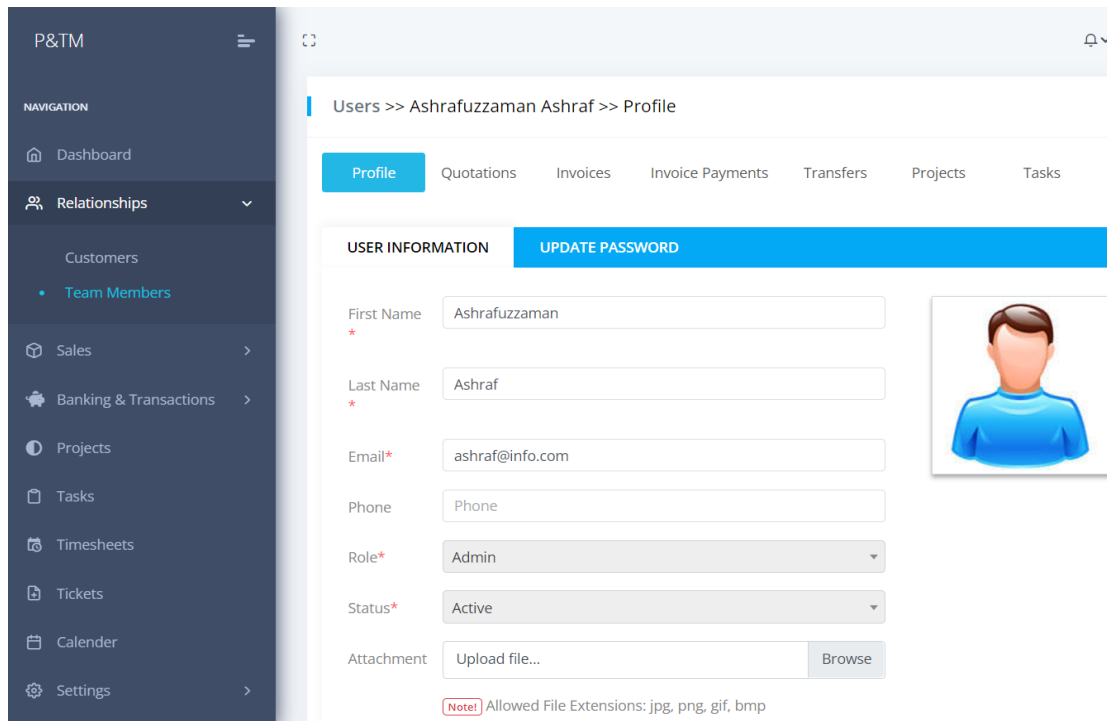


Figure 5.16: Home Page of the Admin.

The Admin Home page is shown in Figure 5.16. An administrator has access to more pages such as Profile, Quotations, Invoices, Invoice Payments, Transfers, Projects, and Tasks. An authentic system user can control the entire web site from this page; additionally, an admin must register from the web sites page. First, we see a list of clients who have signed up for this application.

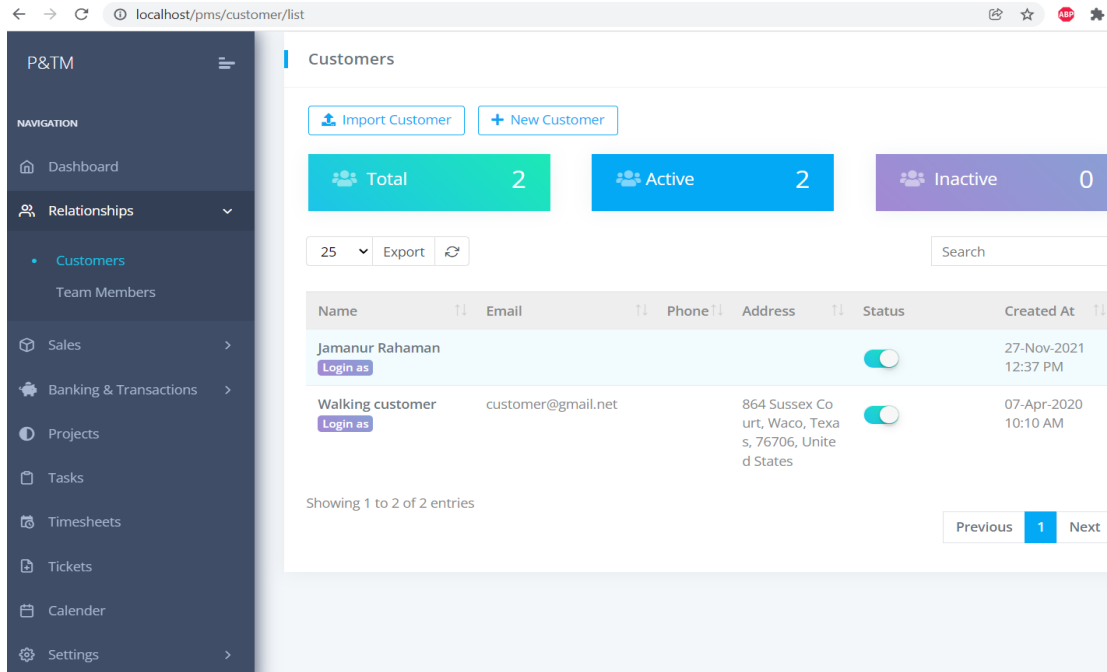


Figure 5.17: Customer List Page

Figure 5.17 depicts a customer list of people who have already registered to receive service. From the admin customer page, admin can add, update, and delete customers. Additionally, the administrator has access to the names and emails of consumers.

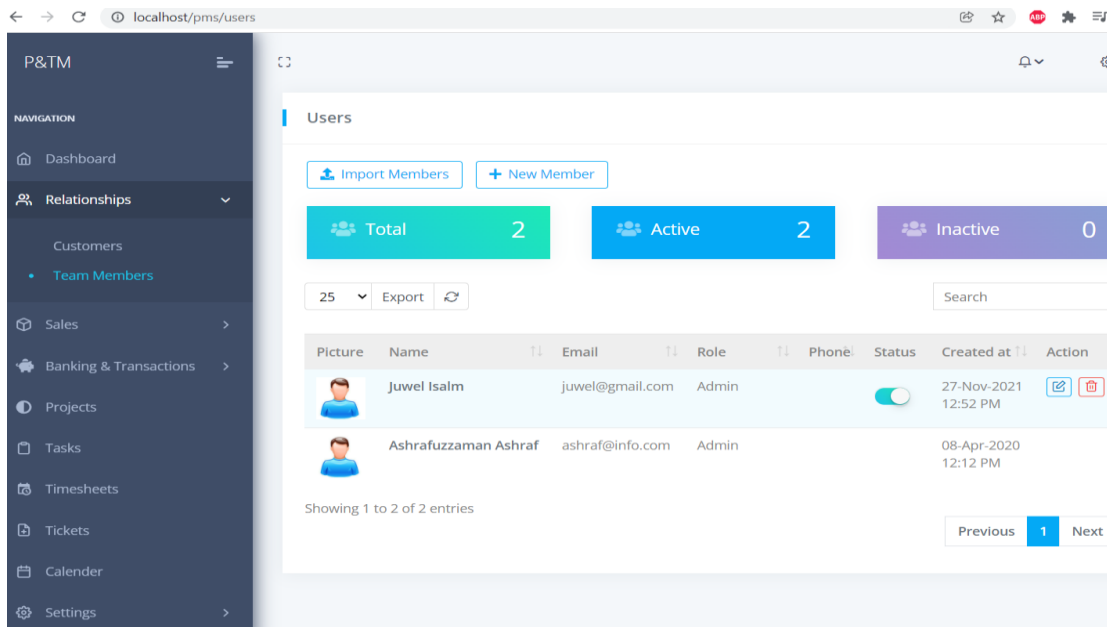


Figure 5.18: Member List of the Admin

Figure 5.18 provides a list of team members who provide customer assistance. The admin team member page allows admin to add, update, and delete team members. In addition, the admin has access to the names and emails of team members.

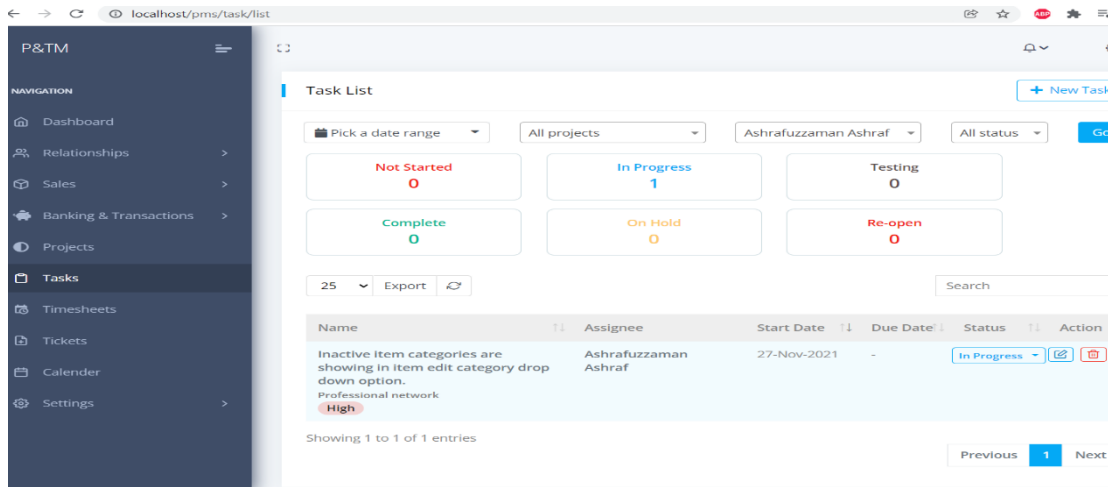


Figure 5.19: Task list for Admin

An admin could accept tasks from this list, as shown in Figure 5.19. The DELETE action allows the administrator to remove tasks from this list. A list of tasks made for clients is displayed at random. This table from the database contains all of the information from customers.

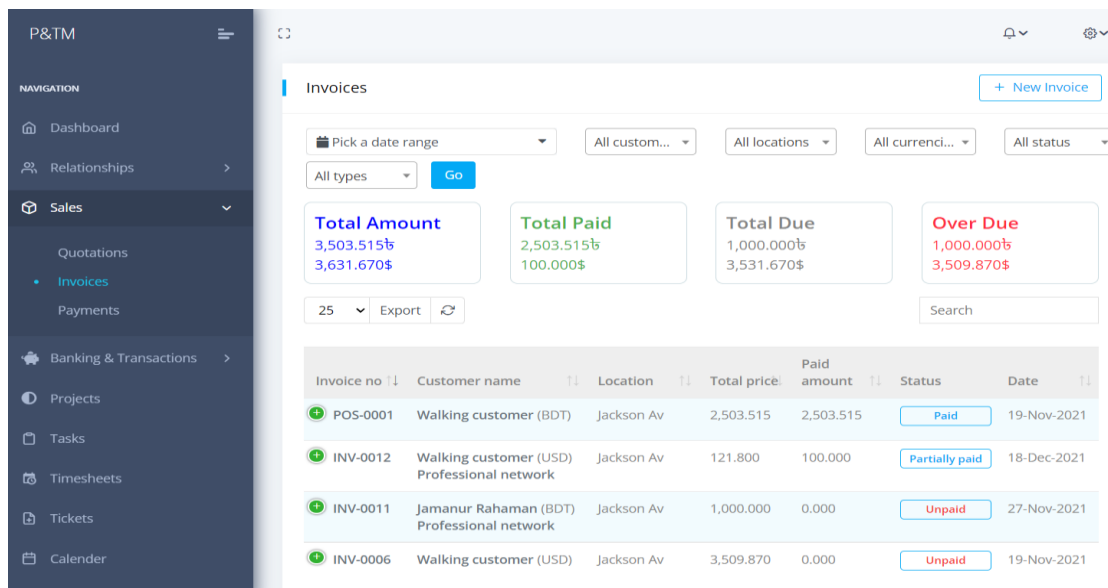


Figure 5.20: Admin Invoice List

All client's invoices can be displayed by an admin in Figure 5.20. Admin can also look at the specifics of your invoice. To alter invoice data or cancel an invoice, select Edit and Delete from this menu.

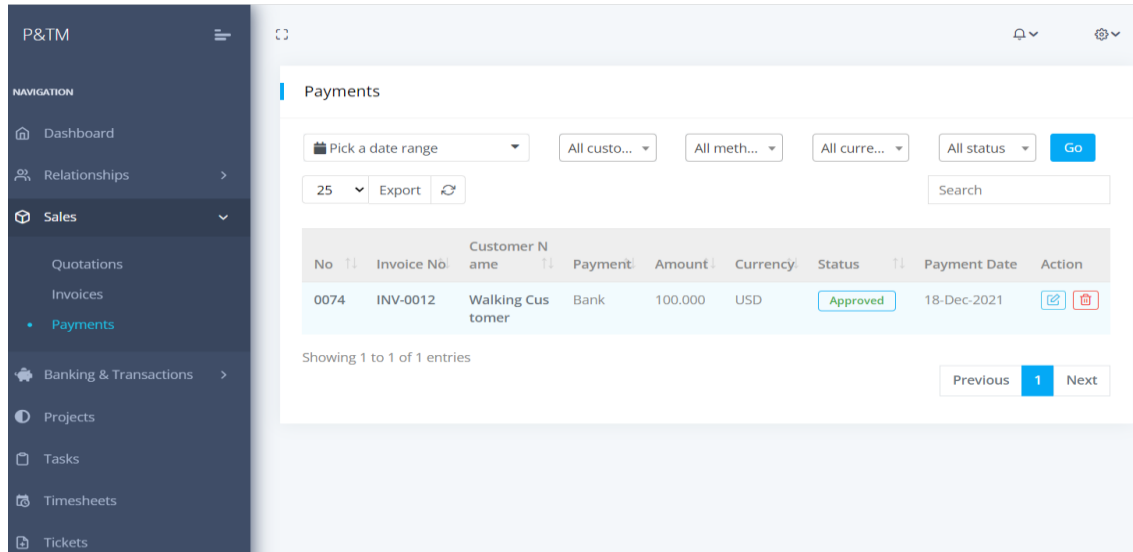


Figure 5.21: Payment List for Admins

Customer's payment lists are depicted in Figure 5.21. Which can be obtained from a reliable source. Admin users can use this page to see which customers have paid which invoice amounts. The administrator has the ability to add, edit, and delete payment actions.

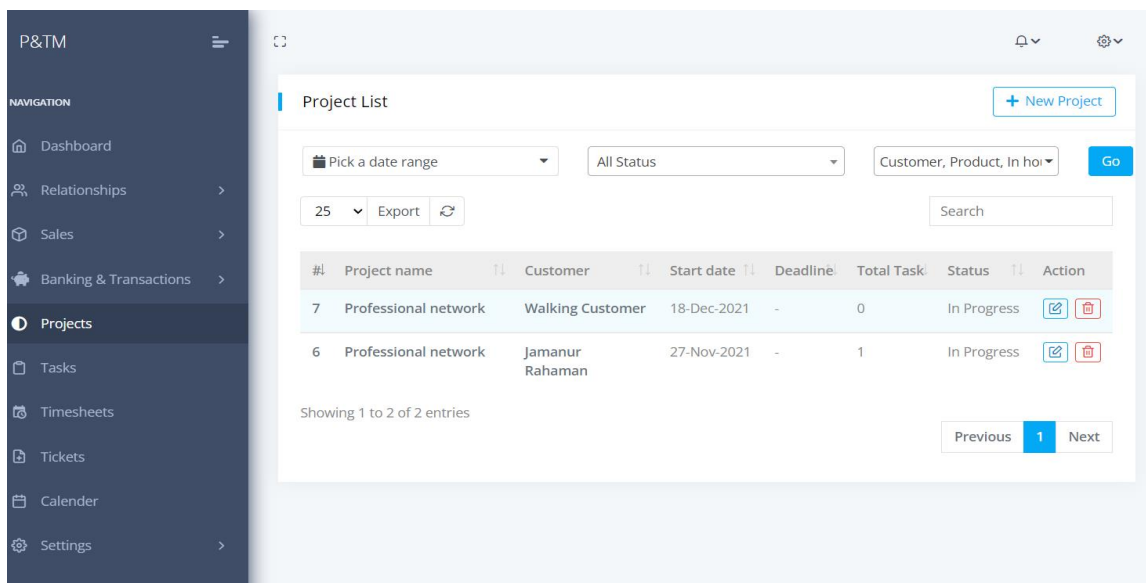


Figure 5.22: List of Admin Projects

The list of project lists that are developed for customers is shown in Figure 5.22. Admin has the ability to Edit and Delete projects. Admin updated the project's pricing, duration, and description. In addition, team members can be assigned to projects.

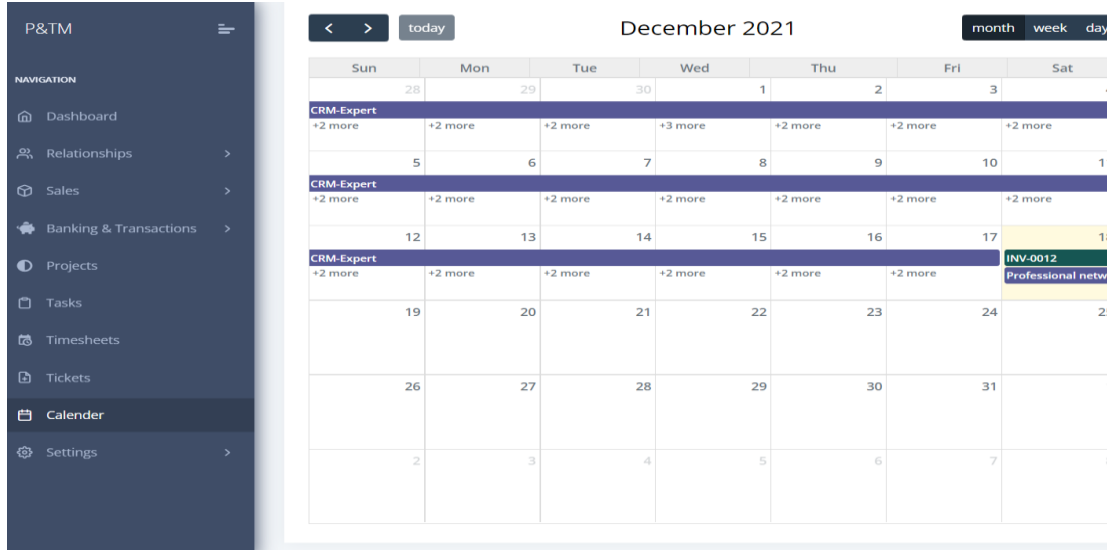


Figure 5.23: Admin Event Calendar

Figure 5.23 depicts the page on which an administrator can examine a calendar by date, week, or month.

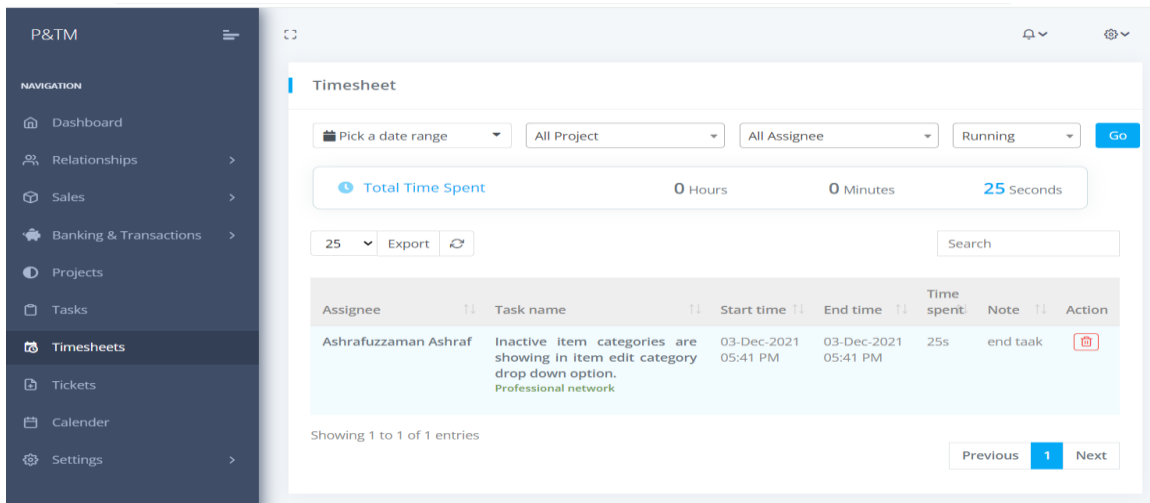


Figure 5.24: Timesheets for Admins.

I depicted the team members' working time sheet in Figure 5.24. The administrator can filter time spent by project, assign members, and status. This time sheet can be edited or deleted by the administrator.

### 5.3 Implementation of Interaction

Interactive applications are critical to the success of any project. As a result, project success hinges on it. Every internet user in our country expects user-friendly software. People are uninterested in any application with a high level of complexity. As a result, my engagement was given top importance in the development of a user-friendly application. So, I did my best to make this web application as user-friendly as possible.

### 5.4 Testing Implementation

I try to test this web application in a variety of ways after it has been successfully launched. Both the user and admin modules test my entire program. Given the fact that some of the examples below:

Table 5.1: Date and details of the test

Test Case	Test Input	Expected Outcome	Obtained Outcome	Result	Tested on
1. User Registration for Customers	Fill out the form completely.	Registration of Customer was successful	Registration was completed successfully.	Pass	07/11/2021

2. Customers User Registration	Your email section is blank.	Alert  Please provide your email address.	Please fill out all of the fields on the form.	Pass	07/11/2021
3. Customers Login	Please enter your proper password to log in.	Alert, please provide a valid password.	There is an issue with logging in.	Pass	08/11/2021
3. Customers Login	Please provide your email address as well as your password.	Customer login has been successful.	"Log In Successfully" appears as an alert message.	Pass	09/11/2021
4.Support ticket	Enter your text here	Successfully send your support ticket	Successfully send your support ticket	Pass	09/11/2021
5.Customer Project Check	Click a Specific Project	Successfully Show project Details	Successfully Show project Details	Pass	09/11/2021



5. Customer Payment	No Enter Payment Amount	Payment Successful	Alert message showed "Enter your Amount"	Pass	09/11/2021
6. Customer Payment	Your given all require info is correct	Your Payment is successful	Alert message showed Payment Successful	Pass	09/11/2021
7.Admin Log In	Incorrect information	Your Access is denied.	Your Access is denied	Pass	10/11/2021
8.Admin Log In	Please Enter your correct information	You have access to log in	Alert message show Admin Login Successful	Pass	10/11/2021
9.Admin Add/Edit Customer	Fill up Field	Your information is correct	Customer Create/Update Successfully	Pass	10/11/2021
10. Admin Add/Edit Team member	Fill up Field	Your information is correct	Member Create/Update Successfully	Pass	10/11/2021
11. Admin Add/Edit Project	Fill up Field	Your information is correct	Project Create/Update Successfully	Pass	11/11/2021

12. Admin Add/Edit Task	Fill up Field	Your information is correct	Task Create/Update Successfully	Pass	11/11/2021
13. Assign Task by admin	Choose Team members.	Member Assign successful	Alert message showed successfully Assign	Pass	11/11/2021
14. Delete task by admin	Select delete option for remove task	Task deleted Successfully	Alert message showed deleted and remove this from list	Pass	13/11/2021
15. Admin replies ticket	Enter text	Send text and update the status	Alert message showed Successfully send	Pass	13/11/2021
16. Admin Add/Edit Payment	Fill up Field	Your information is correct	Payment Create/Update Successfully	Pass	15/11/2021
17. Log Out	Click logout button	Log out the from the profile	Alert message showed Log out successful	Pass	17/11/2021

## **5.5 Testing Result and Reports**

We must test the application in order to detect bugs. During the testing, I encountered a number of bugs. I've also corrected almost all of the bugs I've encountered. More testing of my application is required before I can eliminate more complications. Which are depicted in Table 5.1.

## CHAPTER 6

### CONCLUSION AND FUTURE SCOPE

#### 6.1 Discussion and Conclusion

Any software company will benefit from the Customer Project and Task Management System web application. This online application will help both users and administrators. The complexity of running a business will be reduced for both the owner and the customer. They will gain from the money and timing constraints. In the future, we will introduce more features based on the needs of our clients and business owners.

#### 6.2 Scope for Further Developments

Any development project will continue to develop indefinitely. As a result, I'll need to continue working on this project in the future. Below are a few examples of plans:

- Project deadline and task deadline reminder sent in admin and assigned member via SMS.
- Customer payment and ticket system improve.
- Team member attendance can be added to this web site.
- It is possible to add more functions.
- Security systems can be developed by upgrading user and admin log in systems.

## REFERENCES

- [1] Perfexcrm.com website, <https://perfexcrm.com/>, last accessed on 03 November, 2021
- [2] Rise.fairsketch.com, <https://rise.fairsketch.com/>, last accessed on 06 December, 2021
- [3] Taskhub.company website, <https://taskhub.company/auth>, last accessed on 06 December, 2021
- [4] Crm.infyom.com website <https://crm.infyom.com/login>, last accessed on 06 December, 2021
- [5] Peopleprohrm.com website, <https://peopleprohrm.com/demo/login>, last accessed on 05 December, 2021.
- [6] High.waptechy.com website, <https://high.waptechy.com/>, last accessed on 05 December, 2021.
- [7] Use case diagram tutorial, <http://creatly.com/blog/diagrams/use-case-diagram-tutorial/amp/>, last accessed on 14 December, 2021
- [8] Logical Data Model, <http://www.sciencedirect.com/topics/computer-science/logical-data-model>, last accessed on 14 December, 2021
- [9] Laravel framework, <https://laravel.com/docs/8.x/readme>, last accessed on 10 November, 2021.
- [10] MySQL Storage, <https://www.w3resource.com/mysql/mysql-storage-engines.php> last accessed on 10 November 2021.
- [11] How to Storage, <http://zetcode.com/mysql/storageengines/>, last accessed on 10 November, 2021.
- [13] Image Slider, <https://demonuts.com/image-slider-slideshow/>, last accessed on 14 November, 2021
- [15] Connect, <https://laravel.com/docs/4.2/database>, last accesses on 16 November, 2021

## **APPENDIX**

### **Appendix A: Project Reflection**

I started working on this project in the fall of 2021. For this assignment, I performed a lot of research. I attempted to learn about client projects and task management solutions. I gathered a lot of information from a Bangladeshi software firm. Then I decided to start working on this project. Then I decided on MySQL as the database because it is a simple database that is stable for any administrator. Which can assist the authority in keeping an administrator for this website.

It's difficult to keep it user-friendly. For this website, I created a user-friendly interface. Customers will be able to rely on this interface. As a result, I'm hoping it will simplify things for both corporate administrators and customers. It will be extremely beneficial to customers after the development of new features and facilities. It will bridge the communication gap between the company's administrative staff and its customers. As a result, all customers and owners will be interested in Web-based Customer Project and Task Management System solutions.

# PLAGIARISM

## Plagiarism Report

