CHAPTER 1

1.1 Introduction

The purpose of this project is to create “Smart Pigmy Collection” in this project pigmy collector can login using username and password after login pigmy collector can add the customer and add the daily pigmy payment in system, customer can login to this system and check the deposited payment history in bank login banks can also see all customers deposit payment details.

To implement this project we are going to use full stack development languages like HTML, CSS, JavaScript, Bootstrap, PHP, MYSQL database.

1.2 Capstone Project Scope Document

Capstone Project Title : “Smart Pigmy Collection”

Problem Statement:

As per my survey in current market to collect the payment pigmy collector have been used small machine or manually, to take the machine more payment is required and wastage of printout pages.

Objectives:

Save Time.

Reduce Manual work.

Reduce Machine Cost.

Get easy report to customer.

Capstone project description:

The Purpose of this project is create to "Smart Pigmy Collection" in this project pigmy collector can login using username and password after login pigmy collection can add the customer and add the daily pigmy payment in system customer can login to this system and check the deposited payment history in bank login banks also see the all customer deposit payment details to implement this project we are going to use full stack development languages like HTML, CSS, JavaScript, Bootstrap, PHP, MYSQL database.

Capstone project Deliverables:

Admin/Bank Login
Manage Pigmy Collection
View Pigmy Collection
View Customer
View Payment
Pigmy Collector Login
Add Payment
Add Customer
Customer Login
View Payment

Key Milestones:

Bank Modules

Admin/Bank Login
Manage Pigmy Collection
View Pigmy Collection
View Customer
View Payment

Pigmy Collection Modules

Pigmy Collector Login
Add Payment
Add Customer

Customer Modules

Customer Login

View Payment

Constraints:

Time line for project is limited.

To acquire knowledgeable and skilled labour.

Cost is limited.

Online payment process is not possible because for this we require bank API'S.

Estimated Capstone Project Duration: 72 days(12 weeks)

Estimated Capstone Project Cost: 48,600

Table 1.2.1

Title	**Cost(Rupees)**
Labor Cost	25,410
Cost of material	8,500
Net Profit	14,690
Total	48,600

CHAPTER 2

2.1 Capstone project planning

2.1.1 Work Breakdown Structure

In this project there are three main modules :- Admin modules, Pigmy Collection modules, Customer modules.

Admin/Bank Modules

Admin/Bank Login - In this module using username and password bank can login.

Manage Pigmy Collection - In this module bank can manage the pigmy collection details.

View Pigmy Collection - In this module bank can view the pigmy collection details.

View Customer - In this module bank can view the customer details.

View Payment - In this module bank can view the payment using username and password.

Pigmy Collection Modules

Pigmy collector login - In this module using username and password pigmy collector can login.

Add Payment - In this module pigmy collector can add the customer payment.

Add Customer - In this module using username and password we can login to the portal.

Customer module

Customer login - In this module using username and password customer can login.

View Payment - In this module customer can view the payment using username and password.

Activities And Tasks:

Admin/Bank Login

Tasks

Admin Login must design properly. Entered details must validate properly. Entered detail must be store in database.

Manage Pigmy Collection

Tasks

Bank can manage pigmy collection. Bank must store data properly.

View Pigmy Collection

Tasks

Bank can manage the pigmy collection. Customer can view the collection.

View Customer

Tasks

Bank can view the customer details. Bank must store customer data securely.

View Payment

Tasks

Bank can view payment anytime. Details must be store securely in database.

Pigmy Collection Login

Tasks

Pigmy collection must login using their register details. Register details must validate.

Add Payment

Tasks

Customer can add the payment using valid details. Added payment details must store in database.

Add Customer

Tasks

Pigmy Collector add the customer. Customer details must store to database.

Customer Login

Tasks

Customer can login using their proper details. Entered details must validate.

View Payment

Tasks

Customer can view payment anytime. Details must be store securely in database.

2.1.2 Time-line Schedule

Activities & Task

Admin/Bank Login (7 Days/49 hours)

Tasks

Admin Login must design properly. Entered details must validate properly. Entered detail must be store in database.

Collect requirement gathering and requirement analysis for above tasks.

To design bank login form using HTML, CSS, Bootstrap.

Validation of bank login form using JavaScript.

Perform database connection using PHP and MYSQL.

Testing bank login form to check validation and database.

This whole task is done by Ahmed Peerzade.

Manage Pigmy Collection(6 Days/42 hours)

Tasks

Bank can manage pigmy collection. Bank must store data properly.

Collect requirement gathering and requirement analysis for above tasks.

To design Manage Pigmy Collection form using HTML, CSS, Bootstrap.

Validation of manage pigmy collection form using JavaScript.

Perform database connection using PHP and MYSQL.

Testing Manage Pigmy collection form to check validation and database.

This whole task is done by Ahmed Peerazade.

View Pigmy Collection (7 Days/49 hours)

Tasks

Bank can view the pigmy collection. Customer can view the collection.

Collect requirement gathering and requirement analysis for above tasks.

Perform database connection using PHP and MySQL.

Testing view pigmy collection form to check validation and database.

This whole task is done by Achal Patil.

View Customer (7 Days/49 hours)

Tasks

Bank can view the customer details. Bank must store customer data securely.

Collect requirement gathering and requirement analysis for above tasks.

Perform database connection using PHP and MySQL.

Testing View Customer form to check validation and database.

This whole task is done by Soundarya Bommanhalli.

View Payment(6 Days/42 hours)

Tasks

Bank can view payment anytime. Details must be store securely in database.

Collect requirement gathering and requirement analysis for above tasks.

Perform database connection using PHP and MySQL.

Testing view pigmy collection form to check validation and database.

This whole task is done by Soundarya Bommanhalli.

Pigmy Collector Login (7 Days/49 hours)

Tasks

Pigmy collection must login using their register details. Register details must validate.

Collect requirement gathering and requirement analysis for above tasks.

To design pigmy collection form using HTML, CSS, Bootstrap.

Validation of pigmy collection form using JavaScript.

Perform database connection using PHP and MySQL.

Testing pigmy collection login form to check validation and database.

This whole task is done by Achal Patil.

Add Payment (6 Days/42 hours)

Tasks

Customer can add the payment using valid details. Added payment details must store in database.

Collect requirement gathering and requirement analysis for above tasks.

To design add payment form using HTML, CSS, Bootstrap.

Validation of add payment form using JavaScript.

Perform database connection using PHP and MySQL.

Testing add payment form to check validation and database.

This whole task is done by Soundarya Bommanhalli.

Add Customer (7 Days/49 hours)

Tasks

Pigmy collector add the customer. Customer details must store to database.

Collect requirement gathering and requirement analysis for above tasks.

To design add customer form using HTML, CSS, Bootstrap.

Validation of add customer form using JavaScript.

Perform database connection using PHP and MySQL.

Testing add customer form to check validation and database.

This whole task is done by Achal Patil.

Customer Login (6 Days/42 hours)

Tasks

Customer can login using their proper details. Entered details must validate.

Collect requirement gathering and requirement analysis for above tasks.

To design customer login form using HTML, CSS, Bootstrap.

Validation of customer login form using JavaScript.

Perform database connection using PHP and MySQL.

Testing customer login form to check validation and database.

This whole task is done by Pratibha Jadhav.

View Payment (7 Days/49 hours)

Tasks

Customer can view payment anytime. Details must be store securely in database.

Collect requirement gathering and requirement analysis for above tasks.

To design view payment form using HTML, CSS, Bootstrap.

Validation of view payment form using JavaScript.

Perform database connection using PHP and MySQL.

Testing view payment form to check validation and database.

This whole task is done by Pratibha Jadhav.

2.1.3 Cost Breakdown Structure

Analyse your Work Breakdown Structure

Admin/Bank Login

Manage Pigmy Collection

View Pigmy Collection

View Customer

View Payment

Pigmy Collection Login

Add payment

Add customer

Customer login

View payment

Estimate the cost of materials

The cloud server cost is 5000.

PHP Designer license key of rupees 3450.

XAMPP Server is of free source.

Overhead costs

If we need extra cost for live server.

Build contingency into your CBS

As per the estimated cost we need deliver the project.

Final-check

As per the client budget we have to provide the project budgets.

Table 2.1.3.1

Title	Cost(Rupees)
Labor Cost	25,410
Cost of material	8,500
Net Profit	14,690
Total	48,600

Estimate the labor cost of work

Table 2.1.3.2

Modules	Tasks	Time			Total Cost (hour*amount per hour)
		Hours per task	Hours	Amount per hour	
Bank/Admin login	Admin Login must design properly.	16	49	55	49*55=2,695
	Entered details must validate properly.	16			
	Entered detail must be store in database.	17			
Manage Pigmy Collection	Bank can manage pigmy collection.	21	42	55	42*55=2,310
	Bank must store data properly.	21			
View Pigmy Collection	Bank can view pigmy collection details.	24	49	55	49*55=2,695
	Customer can view the collection.	25			
View Customer	Bank can view the customer details.	24	49	55	49*55=2,695
	Bank must store customer data securely.	25			
View Payment	Bank can view payment anytime.	21	42	55	42*55=2,310
	Details must be store securly in database.	21			
Pigmy Collection Login	Pigmy collection must login using their register details.	24	49	55	49*55=2,695

	Registered details must validate.	25			
Add Payment	Customer can add the payment using valid details.	21	42	55	42*55=2,310
	Added payment details must store in database.	21			
Add Customer	Pigmy collector add the customer.	24	49	55	49*55=2,695
	Customer details must store to database.	25			
Customer Login	Customer can login using their proper details.	21	42	55	42*55=2,310
	Entered details must validate.	21			
View Payment	Customer can view payment anytime.	24	49	55	49*55=2,695
	Details must be store securely in database.	25			

2.1.4 Risks Assessment

Home page must display Admin login, Pigmy Collector Login, Customer Login, View Payment, Contact US, About Us and Home all of these must be displayed and work properly.

Admin login form must work without any error.

Admin must be able to see the number of pigmy collectors and the number of customers, status to activate or inactivate and delete option.

The customer registration must work and be registered without any error.

Customer login form must work without any error.

Customer should add the payment without any error.

Registration list of the customers must be visible.

2.2 Requirements Specification

2.2.1 Functional requirements:

Admin Aspects

Admin login

Manage Pigmy Collector

View Pigmy Collector

View Customer

View Payment

Pigmy Collector Aspects

Pigmy Collector Login

Add Customer

Add Payment

Customer Aspects

Customer Login

View Payment

2.2.2 Non-functional

Usability

Usability is a quality attribute used to access how easy the interface is to use. Usability is ease of use. It tells how user friendly the interface is. It includes memorability, learnability, and satisfaction. Our software interface has all the above quality. Any kind of user can easily understand the interface.

Reliability

Reliability is how much the system is consistent in different platforms. The ability of an apparatus, system to consistently perform its required function, on demand and without degradation or failure.

Integrity

Integrity means doing the right thing in a reliable way. Data integrity is a fundamental component of security. In its broadcast use, “Data Integrity” refers to the accuracy and consistency of data stored in a database, data mart or another construct. Data integrity is the overall completeness, accuracy and consistency of data.

Performance

Performance is also a major non-functional requirement. Performance Requirements about resources required, response time, transaction rate or anything else having to do with performance.

2.2.3 User inputs

Admin:

Username and password

Manage Pigmy Collection

View The Customer

View The Payment

Pigmy Collector Login:

Add The Customer

Add The Payment

Customer Login:

View The Payment

2.2.4 Technical Constraints

Requirements for developing: -

Average I3 processor.

Average 4GB RAM.

Average 512GB hard disk.

Hardware requirement for deployment: -

Average I3 processor.

Average 2GB RAM.

Average 256GB storage.

Software requirement for development: -

OS (Window’s, MAC, Linux).

XAMP Server.

VS Code.

Chrome.

Software requirement for deployment: -

OS.

Chrome.

Language used or technology:-

Content language (client side language) (HTML, CSS, Bootstrap, JavaScript).

Backend (server side language) (PHP).

Database (MySQL)

2.3 Design Specification

2.3.1 Chosen System Design

System architecture diagrams provide a visual illustration of a system's various components and show how they communicate and interact with each other. These diagrams document a system's structure and architecture.

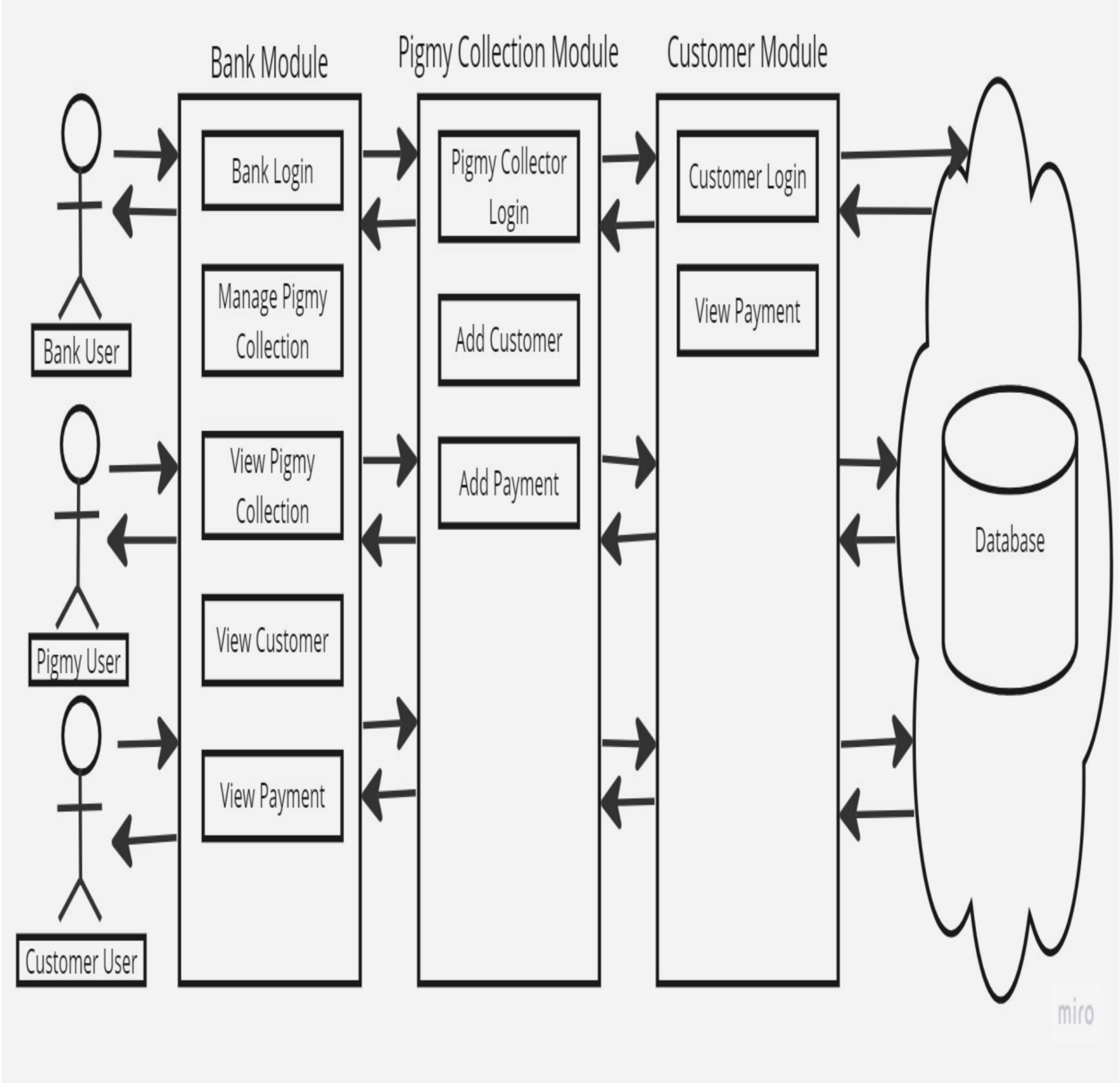


Figure 2.3.1.1

2.3.2 Discussion of alternative designs

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

DFD Level-0

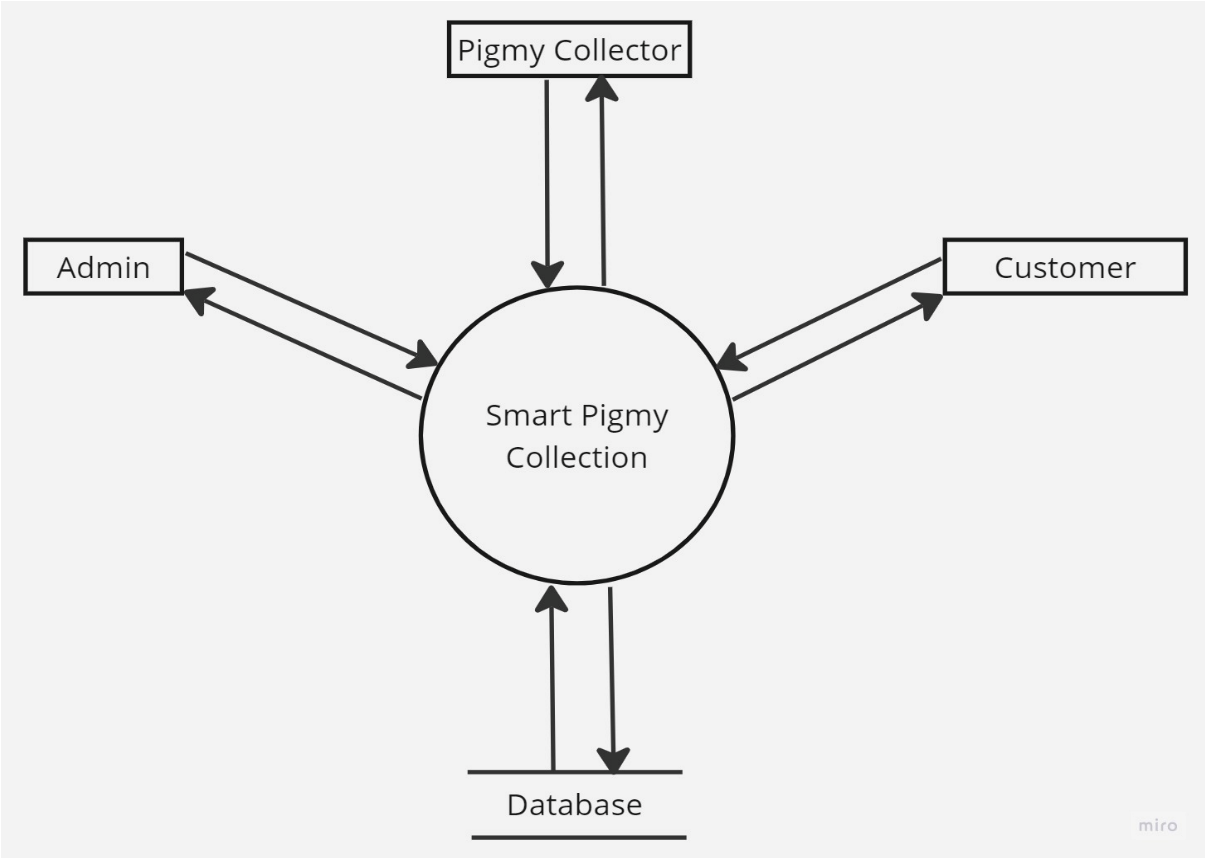


Figure 2.3.2.1

DFD Level-1 Admin

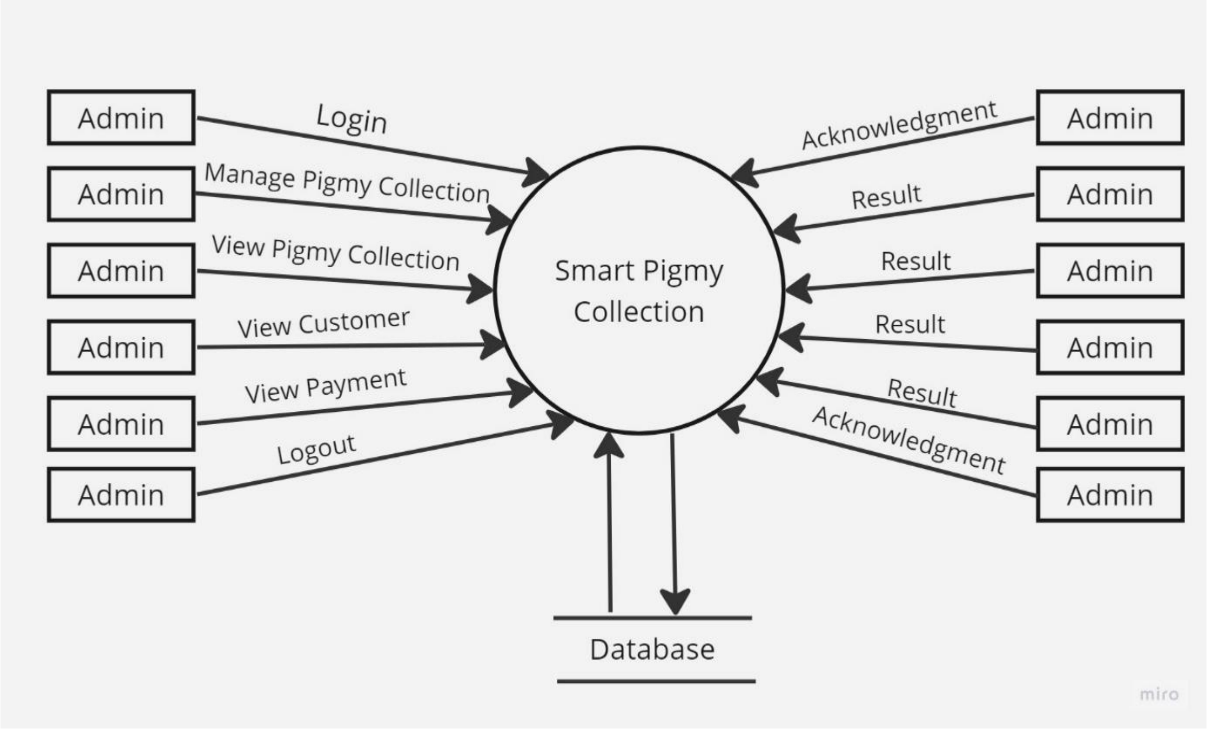


Figure 2.3.2.2

DFD Level-1 Pigmy collector

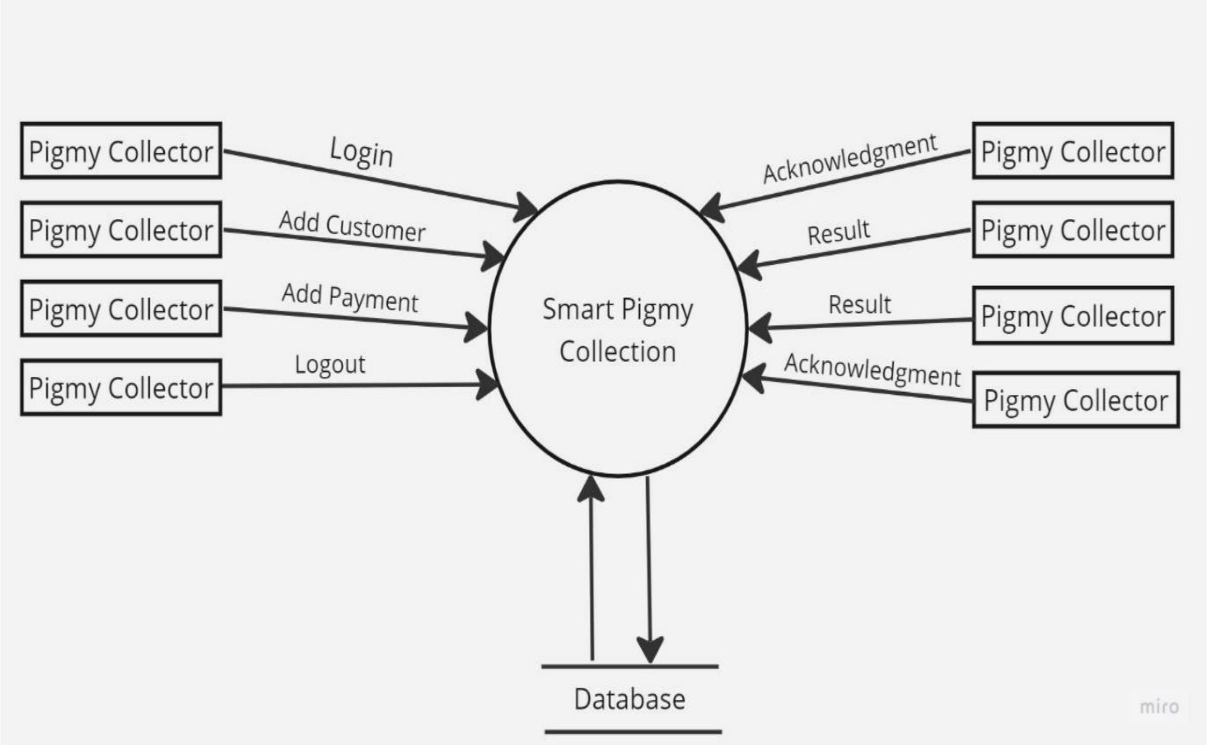


Figure 2.3.2.3

DFD Level-1 Customer

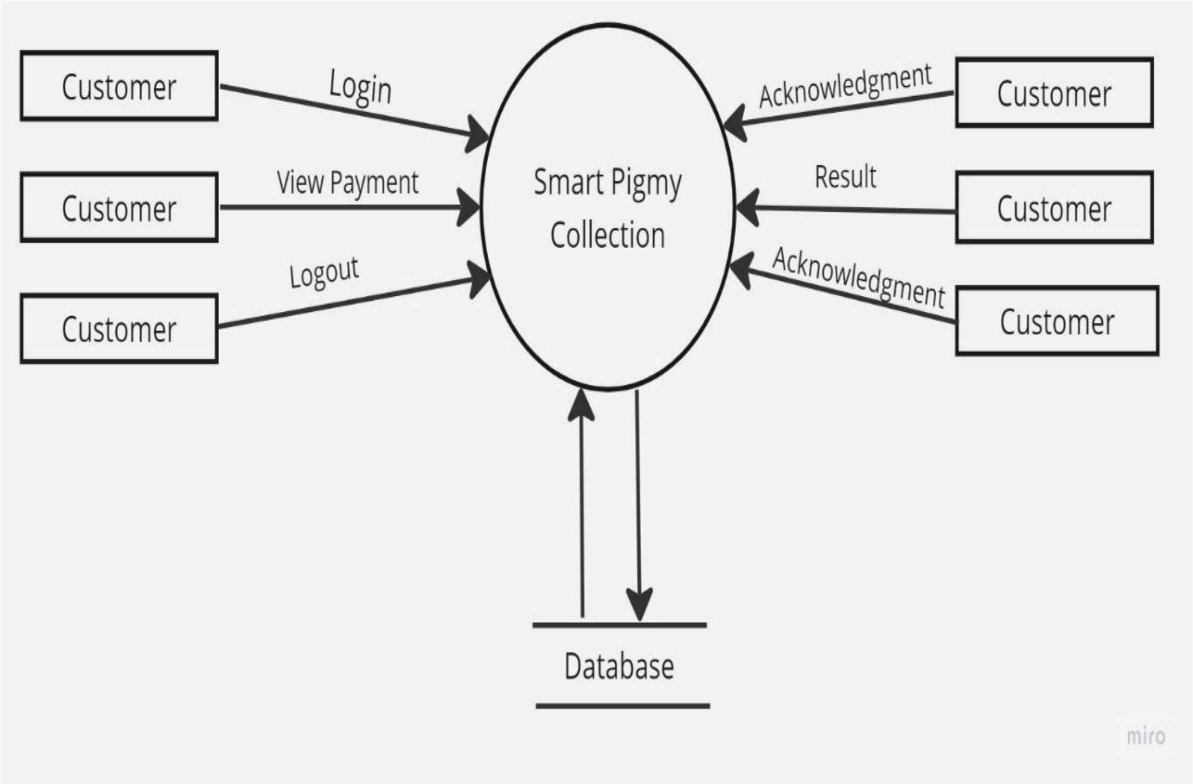


Figure 2.3.2.4

2.3.3 Detailed Description of Components/subsystems

Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

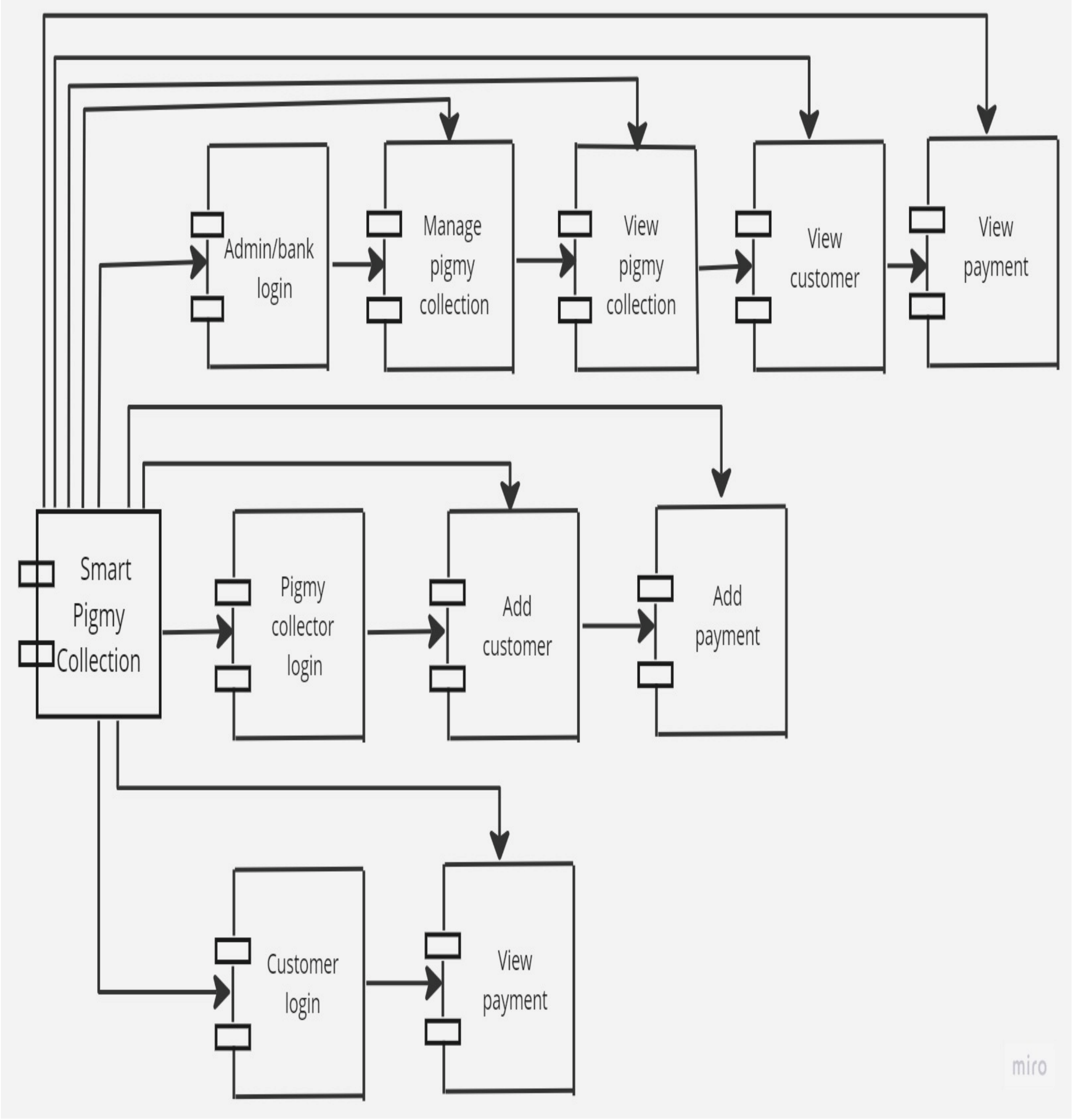


Figure 2.3.3.1

Admin/bank login

Here admin can login to the form using admin username and password. After logging in he can add the pigmy collector, he can add or delete customers or pigmy collectors.

Manage pigmy collection

After admin logged in to the form admin can easily manage the accounts he had added. He can add or manage the pigmy collection.

View pigmy collection

Admin can also manage or view the pigmy collection using username and password.

View customer

Customer can login using their valid details. Also he can view its payment details using username and password.

View payment

Customer can view payment details using their valid details and also admin can view the payment history.

Pigmy collector login

Pigmy collector can login to the form using username and password and he can also add the customers.

Add Customer

Pigmy collector adds the customers using their correct details and he can also update the customers details regularly.

Add payment

Pigmy collector can add payment and payment details using valid details and he can update the payment history.

Customer login

Customer can login to the form using username and password and he can also view the payment details.

View payment

Customer can view payment details using their valid details and also admin can view the payment history.

2.3.4 Component 1-n

Admin/Bank Login

Manage Pigmy Collection

View Pigmy Collection

View Customer

View Payment

Pigmy Collection Login

Add payment

Add customer

Customer login

View payment

CHAPTER 3

3.1 Approach and Methodology

3.1.1 Discuss the technology

Web technology

A place connected to the internet, where a company, organization, etc. Puts information that can be found on the World Wide Web.

Types of web technology: -

Browsers

HTML and CSS

Programming Languages

Frameworks

Web Servers

Databases

Protocols

Lastly, data Formats

Advantages

We can access from any location.

No data loss.

Data can be recovered.

Be available every time.

We can save time.

Cloud based technology

To live the project we need to purchase cloud sever.

Cloud based technology is the use of software and services via the internet .These applications commonly include data storage, networking, servers and databases. User can access their cloud hosted tools with any device that is connected to the internet.

Types of cloud based technology

SAAS: - Software as a Service

PAAS: - Platform as a Service

IAAS: - Infrastructure as a Service

Advantages

Usability and accessibility

Security

Cost efficient Convenient sharing of files

Automation

Open-Source web technology

For our project we not require to purchase any software's and libraries

XAMPP

XAMPP is a cross-platform and open source tool, which makes it an ideal choice of web developers. It is the acronym of X-cross platform, Apache, MySQL, PHP, and Perl.

PHP

PHP is an open source scripting language used for creating dynamic and interactive web pages and various digital platforms.

PhpMyAdmin

PhpMyAdmin is an open source and free administration tool for MySQL

Advantages

Community-Driven Reliability

Community-Driven Security

Low Cost on an Ongoing Basis

Better, Community-Based Collaboration

3.1.2 Methodologies

Agile methodology is a process for managing a project that involves constant collaboration and working in iterations. Agile project management works off the basis that a project can be continuously improved upon throughout its life cycle, with changes being made quickly and responsively. Agile is one of the most popular approach to project Management due to it‘s flexible, adaptability to change, and high level of customer input.

Phases of Agile model

Requirement Gathering
Design the requirements
Construction or Iterations
Testing and Quality assurance
Deployment
Feedback

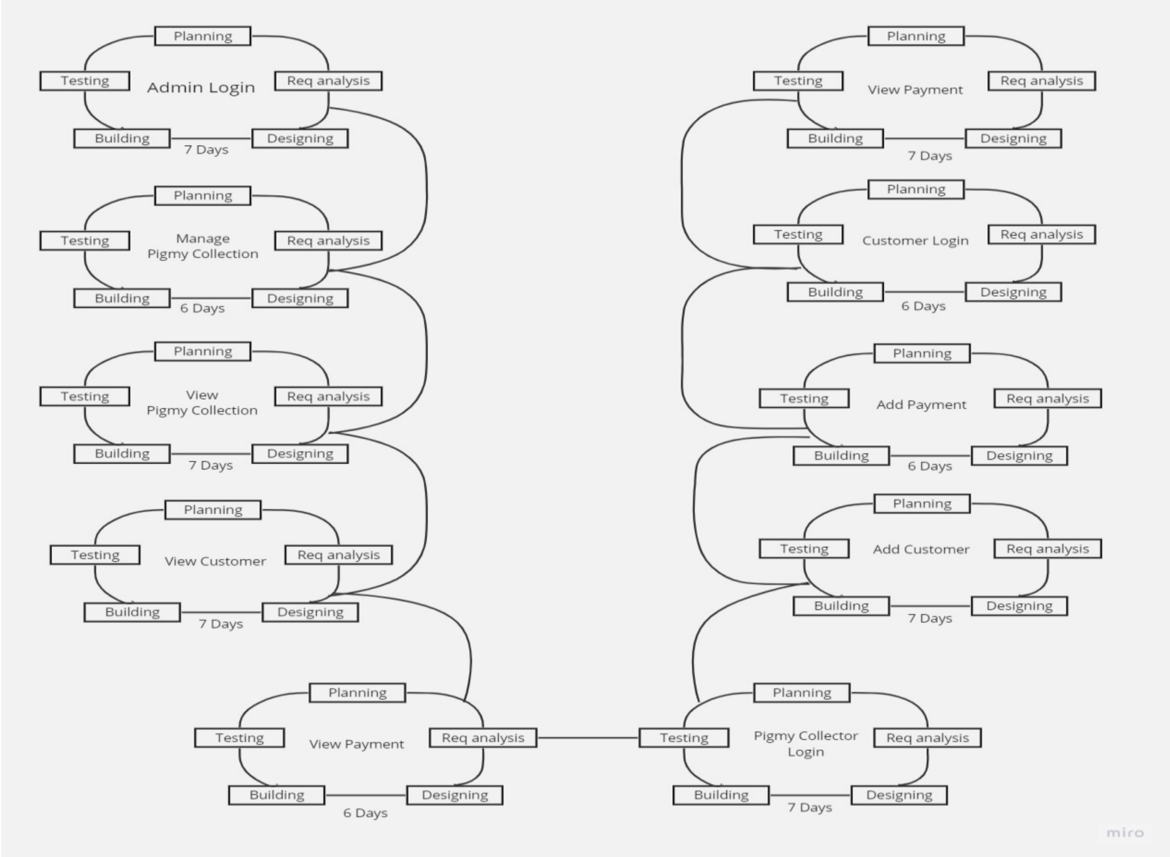


Figure 3.1.2.1

3.1.3 Use Cases

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen.

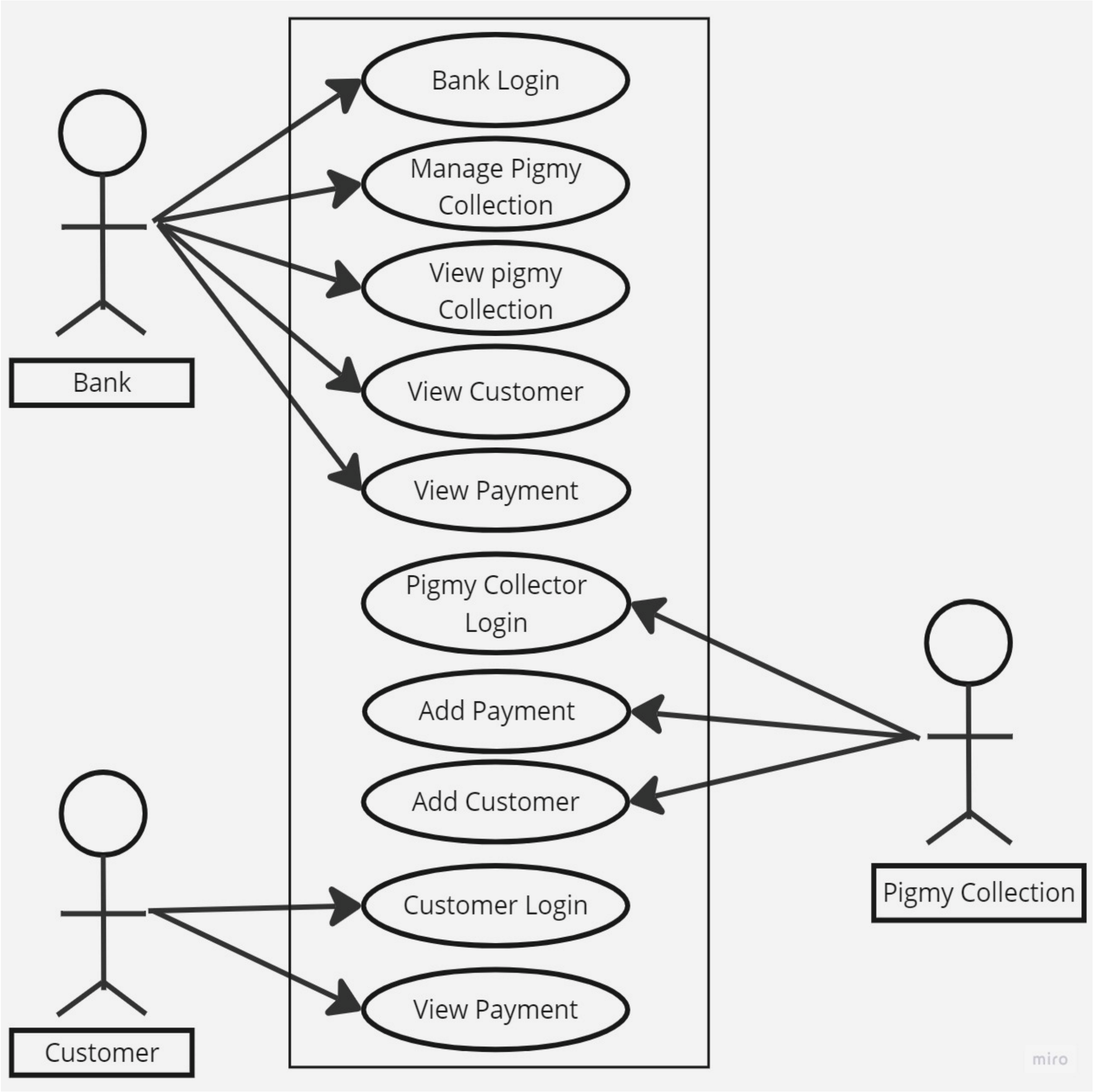


Figure 3.1.3.1

3.1.4 Programming

HTML

The hypertext mark-up language or HTML is the standard mark-up language for documents designed to be displayed in a web browser. It is often assisted by technologies such as CSS and scripting languages such as JavaScript.

Advantages

It is easy to learn.

Every browser supports HTML Language.

HTML is light weighted and fast to load.

CSS

Cascading style sheet is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML.CSS is a cornerstone technology of the World Wide Web.

Advantages

Improve the browsing speed.

It can be used on various devices.

Wider variety of design options.

JavaScript

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

Advantages

Fast speed

Easy to learn

Versatility

Popularity

Bootstrap

Bootstrap is a free and open source CSS framework directed at responsive, front end web development. It contains HTML, CSS and java script based design templates for forms, buttons, navigation, etc.

Advantages

Open source

Easy to use

Save lots of time

Compatible with browsers

PHP

PHP hypertext pre-processor is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML.PHP pages contains HTML with embedded code.

Advantages

It’s open-source and free from cost.

It is platform-independent.

It helps in managing code easily

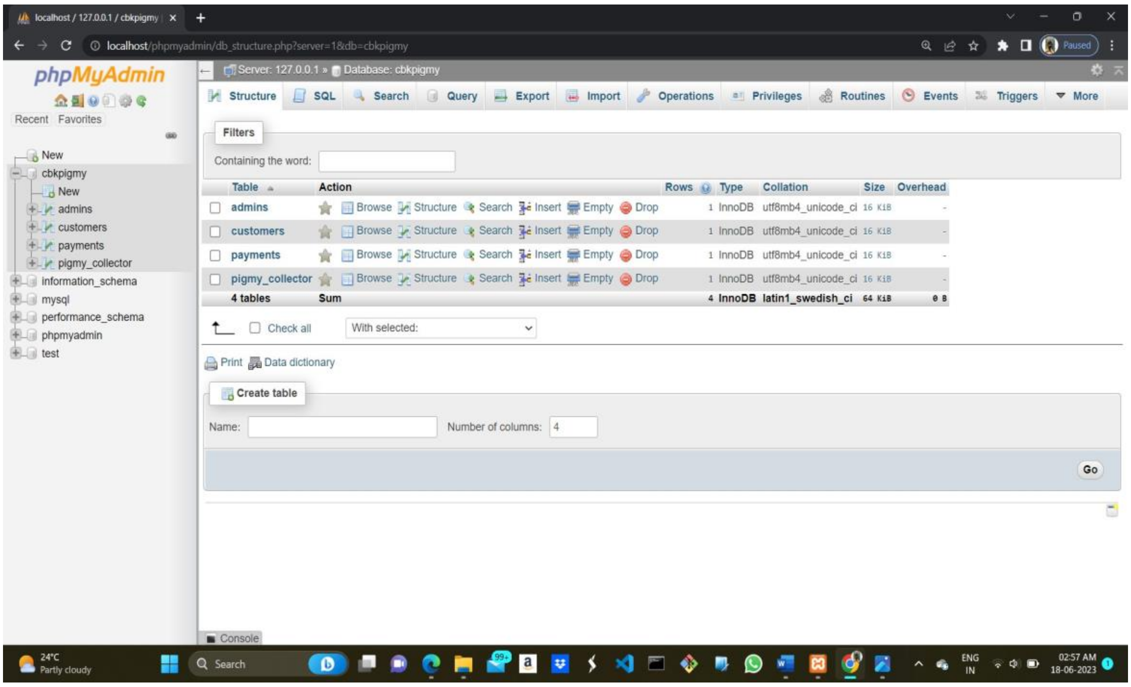
3.1.5 Analysis

As per my survey in current market to collect the payment pigmy collector are used small machine or manually to take the machine more payment is required and wastage of printout pages.

we are going to develop smart pigmy collection can login using username and password after login pigmy collector can add the customer and add the daily pigmy payment in system customer can login to this system and check the deposited payment history in bank login banks also see the all customer deposit payment details to implement this project we’re going to use full stack development languages like HTML, CSS, JavaScript, Bootstrap, PHP, MYSQL database.

3.1.5 Process Design

List of tables



localhost/phpmyadmin/db_structure.php?server=1&db=cbkpigmy

Server: 127.0.0.1 » Database: cbkpigmy

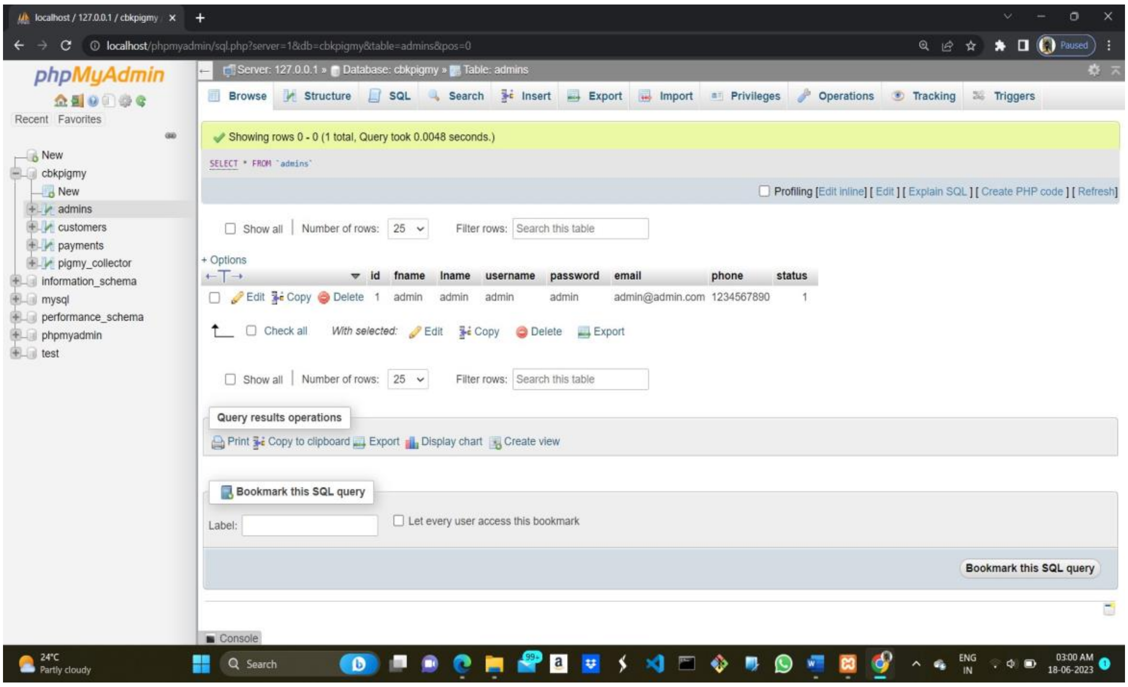
	Table	Action	Rows	Type	Collation	Size	Overhead
□	admins	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	customers	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	payments	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	pigmy_collector	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
	4 tables	Sum	4	InnoDB	latin1_swedish_ci	64 KiB	0 B

Name:

Number of columns: 4

Figure 3.1.6.1

Admin table



localhost/phpmyadmin/sql.php?server=1&db=cbkpigmy&table=admins&pos=0

Server: 127.0.0.1 » Database: cbkpigmy » Table: admins

Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
+ Options
Check all
With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Let every user access this bookmark
Console

✓ Showing rows 0 - 0 (1 total, Query took 0.0048 seconds.)

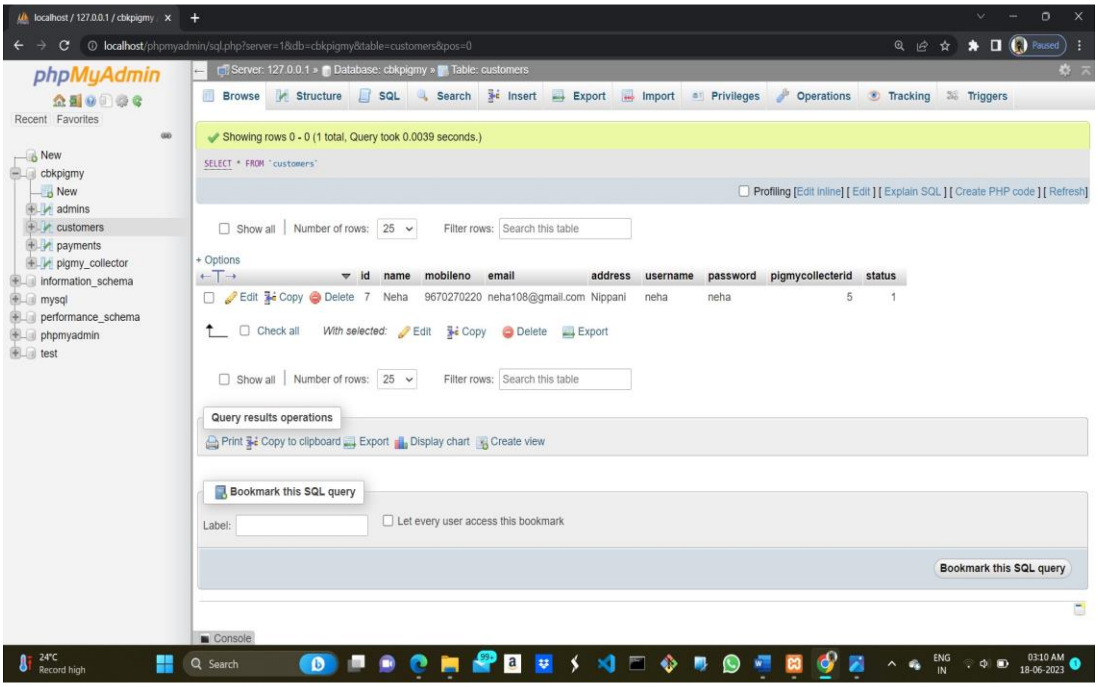
```
SELECT * FROM `admins`
```

		id	fname	lname	username	password	email	phone	status
□	Edit Copy Delete	1	admin	admin	admin	admin	admin@admin.com	1234567890	1

Label:

Figure 3.1.6.2

Customer table



localhost/phpmyadmin/sql.php?server=1&db=cbkpigmy&table=customers&pos=0

Server: 127.0.0.1 » Database: cbkpigmy » Table: customers

✓ Showing rows 0 - 0 (1 total, Query took 0.0039 seconds.)

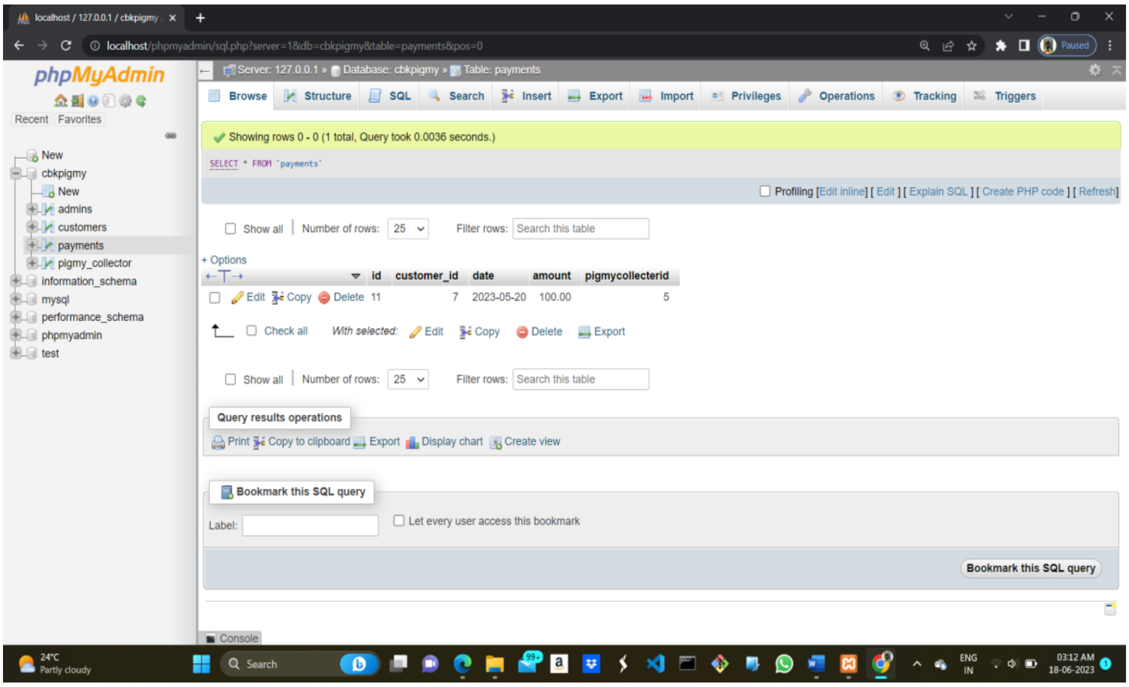
```
SELECT * FROM `customers`
```

id	name	mobileno	email	address	username	password	pigmycollecterid	status
7	Neha	9670270220	neha108@gmail.com	Nippani	neha	neha	5	1

localhost / 127.0.0.1 / cbkpigmy
phpMyAdmin
Recent Favorites
New
cbkpigmy
New
admins
customers
payments
pigmy_collector
information_schema
mysql
performance_schema
phpmyadmin
test
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
+ Options
□ Edit Copy Delete
□ Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
□ Let every user access this bookmark
Bookmark this SQL query
Console
24°C Record high
Search
ENG IN
03:10 AM 18-06-2023

Figure 3.1.6.3

Payment table



localhost/phpmyadmin/sql.php?server=1&db=cbkpigmy&table=payments&pos=0

Server: 127.0.0.1 » Database: cbkpigmy » Table: payments

✓ Showing rows 0 - 0 (1 total, Query took 0.0036 seconds.)

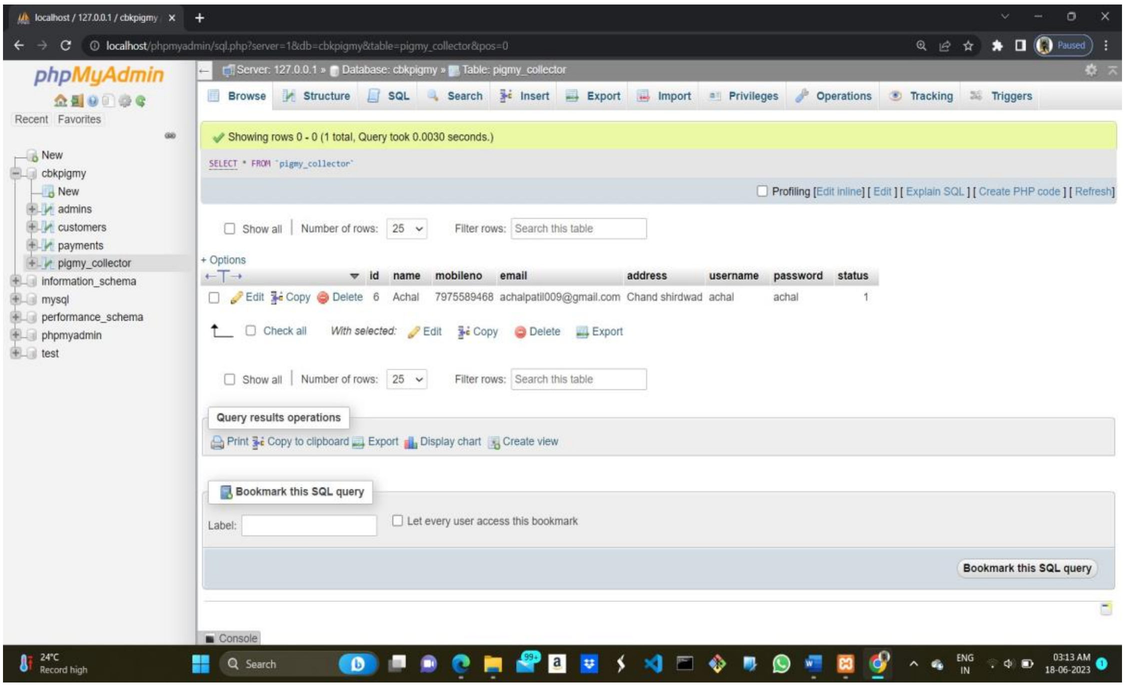
```
SELECT * FROM `payments`
```

id	customer_id	date	amount	pigmycollecterid
11	7	2023-05-20	100.00	5

localhost / 127.0.0.1 / cbkpigmy
phpMyAdmin
Recent Favorites
New
cbkpigmy
New
admins
customers
payments
pigmy_collector
information_schema
mysql
performance_schema
phpmyadmin
test
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
+ Options
□ Edit Copy Delete
□ Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
□ Let every user access this bookmark
Bookmark this SQL query
Console
24°C Partly cloudy
Search
ENG IN
03:12 AM 18-06-2023

Figure 3.1.6.4

Pigmy collector table



localhost / 127.0.0.1 / cbkpigmy
localhost/phpmyadmin/sql.php?server=1&db=cbkpigmy&table=pigmy_collector&pos=0
phpMyAdmin
Recent Favorites
New
cbkpigmy
New
admins
customers
payments
pigmy_collector
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: cbkpigmy » Table: pigmy_collector
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

✓ Showing rows 0 - 0 (1 total, Query took 0.0030 seconds.)

```
SELECT * FROM `pigmy_collector`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
+ Options

	id	name	mobileno	email	address	username	password	status
□ Edit Copy Delete	6	Achal	7975589468	achalpatil009@gmail.com	Chand shirdwad	achal	achal	1

□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label: □ Let every user access this bookmark
Bookmark this SQL query
Console
24°C Record high
Search
ENG IN
03:13 AM 18-06-2023

Figure 3.1.6.5

3.1.6 Product Design

Index Page

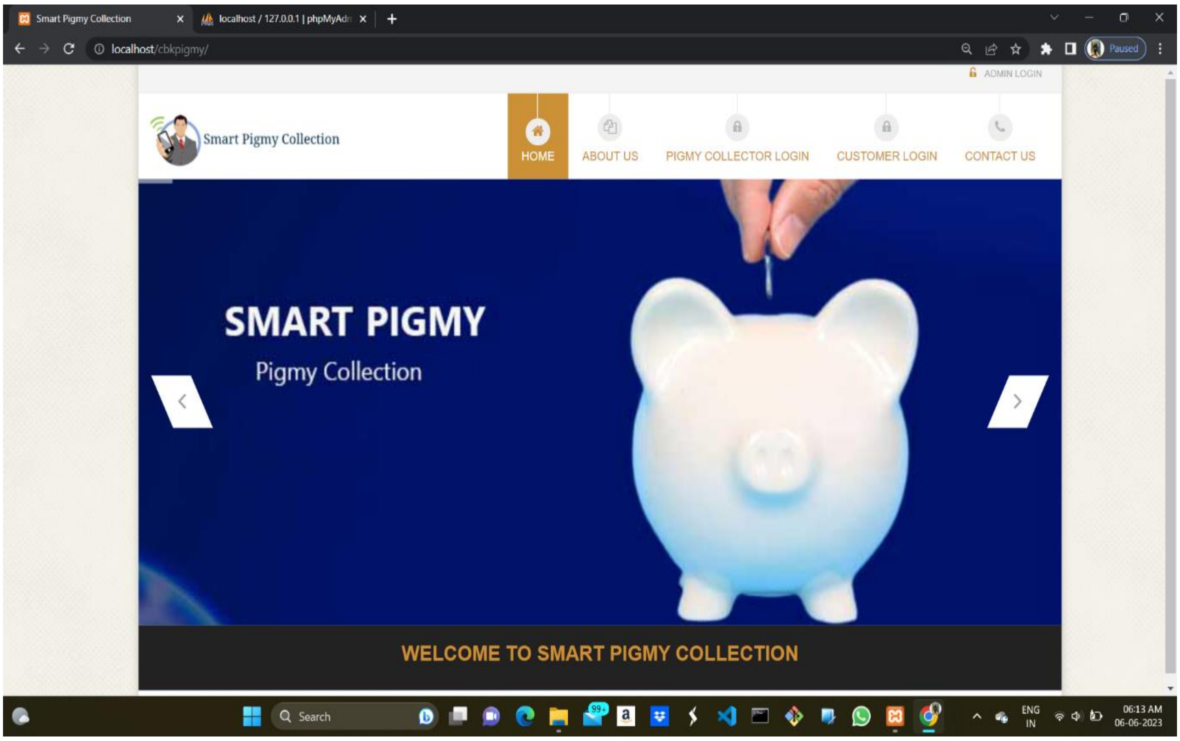
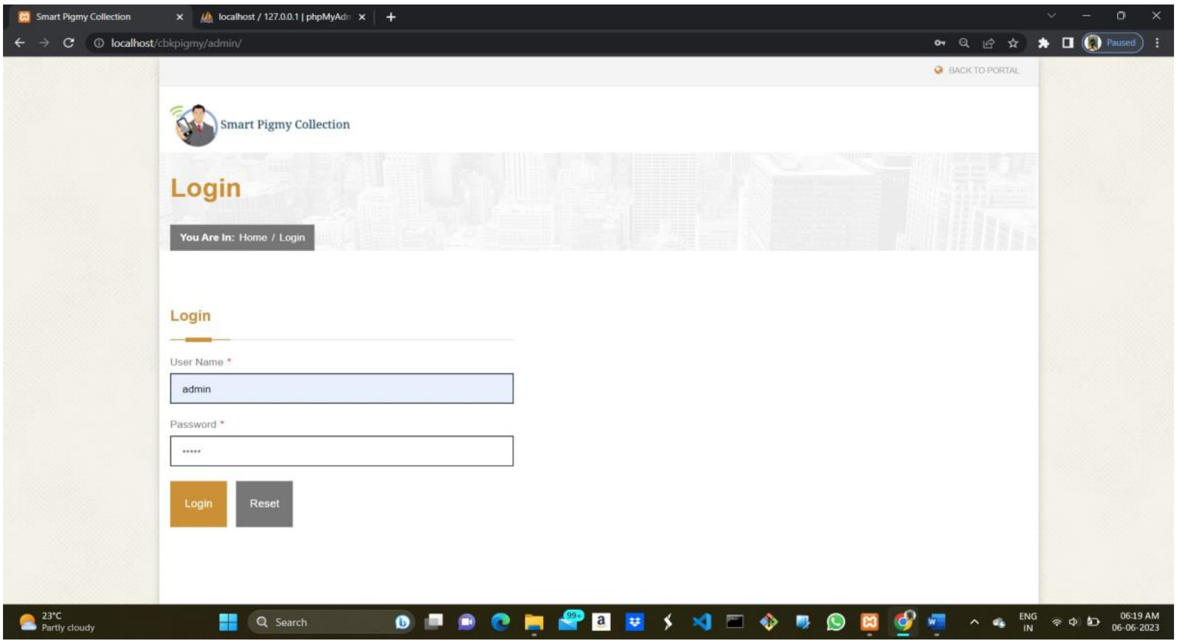


Figure 3.1.7.1

Admin Login Page

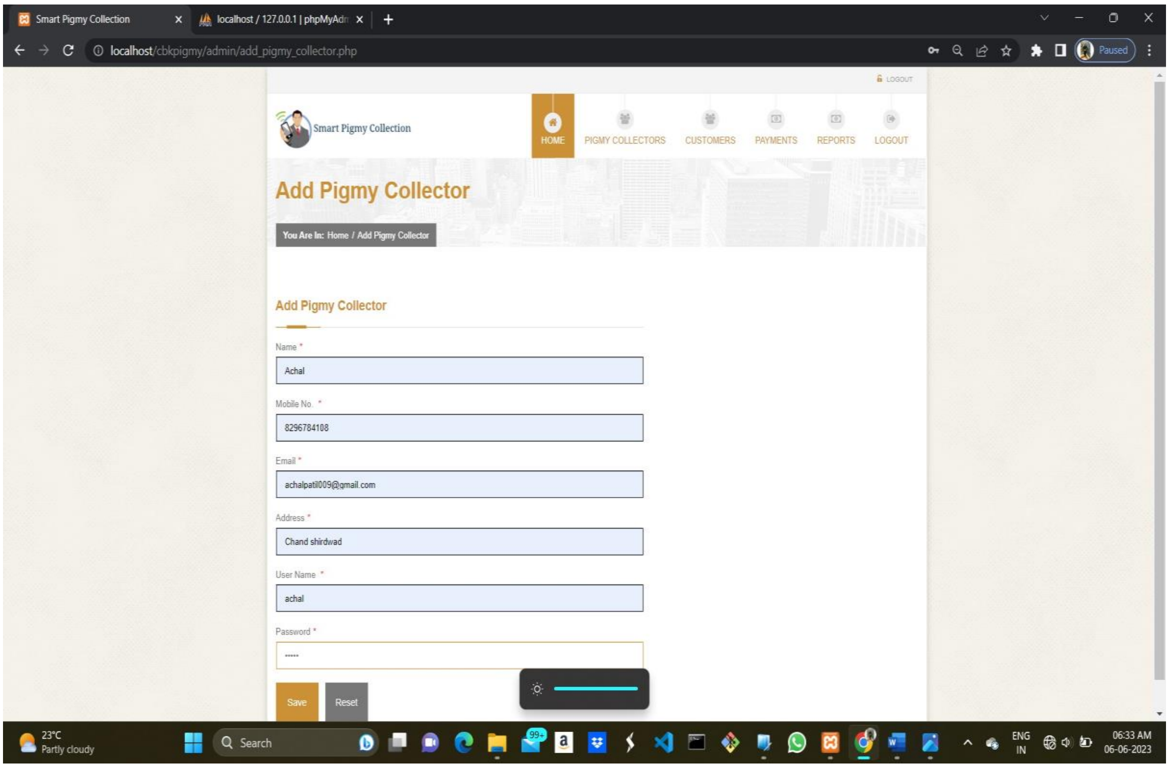


User Name * admin

Password * •••••

Figure 3.1.7.2

Add Pigmy Collector Page



Add Pigmy Collector

You Are In: Home / Add Pigmy Collector

Add Pigmy Collector

Name * Achal

Mobile No. * 8296784108

Email * achalpatil009@gmail.com

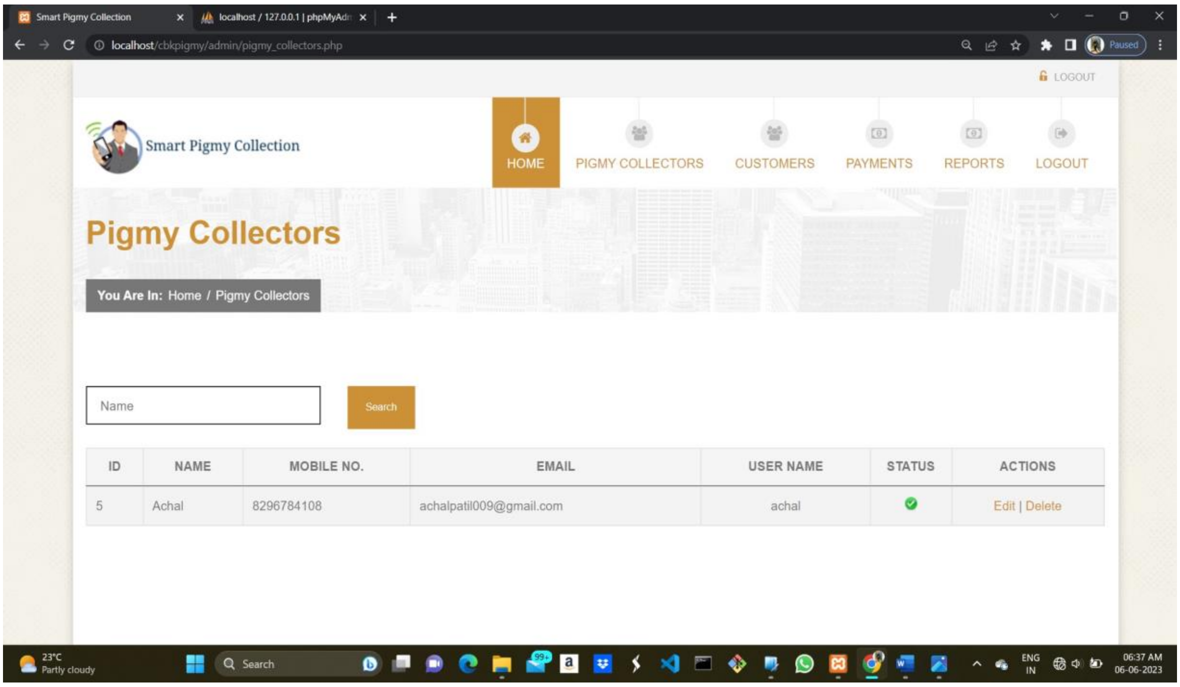
Address * Chand shirdwad

User Name * achal

Password * •••••

Figure 3.1.7.3

View Pigmy Collector Page



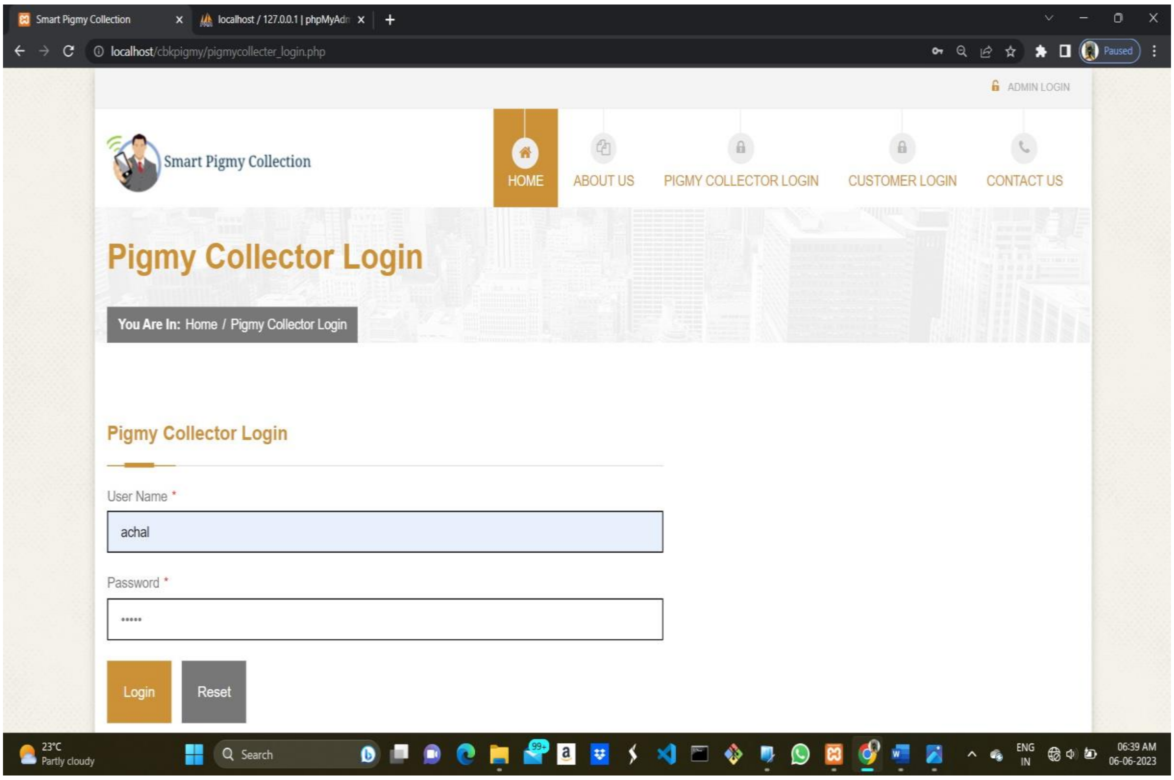
Pigmy Collectors

You Are In: Home / Pigmy Collectors

ID	NAME	MOBILE NO.	EMAIL	USER NAME	STATUS	ACTIONS
5	Achal	8296784108	achalpatil009@gmail.com	achal	✓	Edit | Delete

Figure 3.1.7.4

Pigmy Collector Login Page



Pigmy Collector Login

User Name *

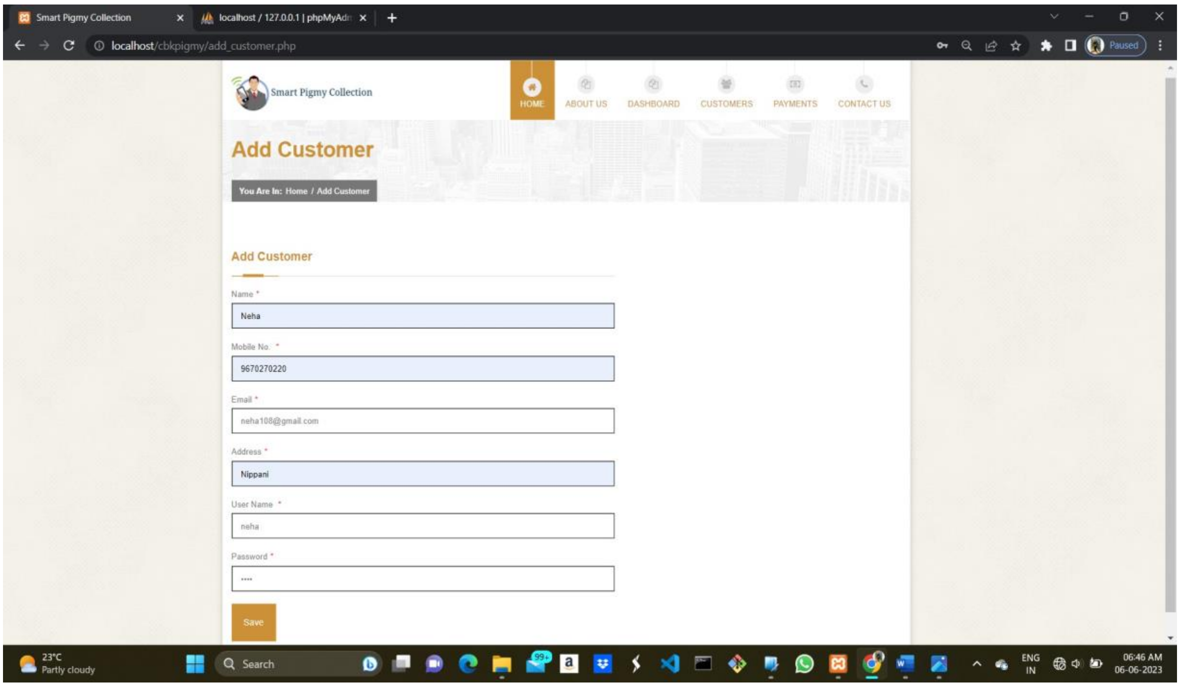
achal

Password *

•••••

Figure 3.1.7.5

Add Customer Page



Add Customer

Name *

Neha

Mobile No. *

9670270220

Email *

neha108@gmail.com

Address *

Nippani

User Name *

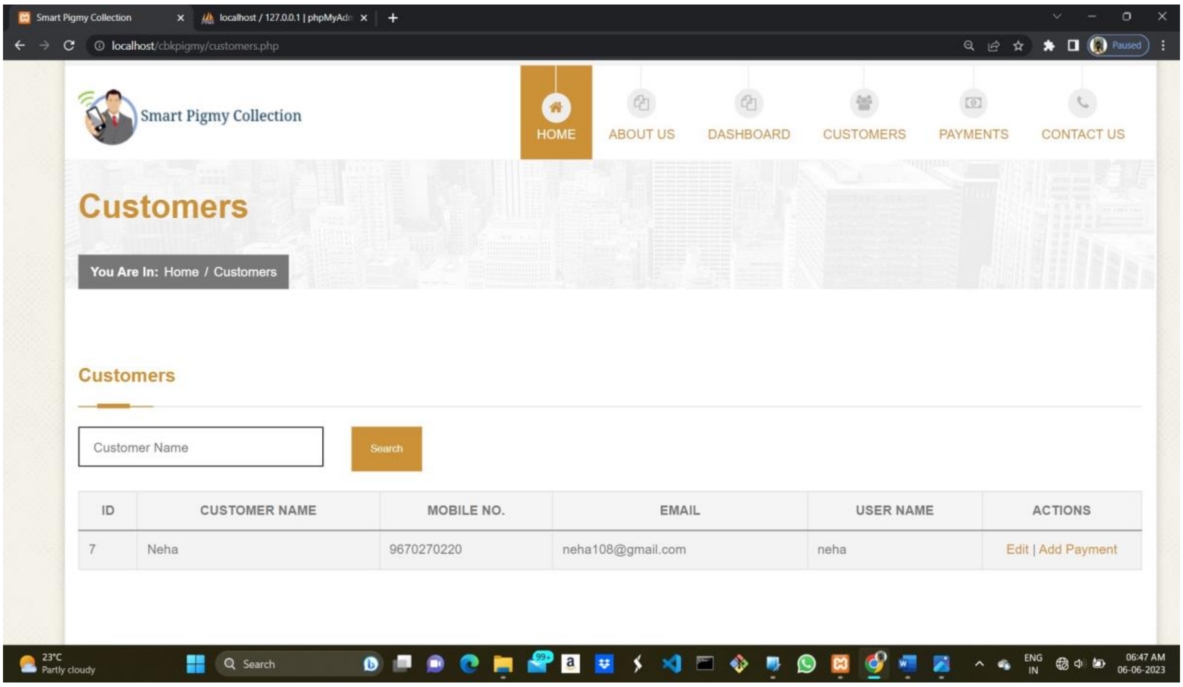
neha

Password *

••••

Figure 3.1.7.6

View Customer Page



Customers

You Are In: Home / Customers

Customers

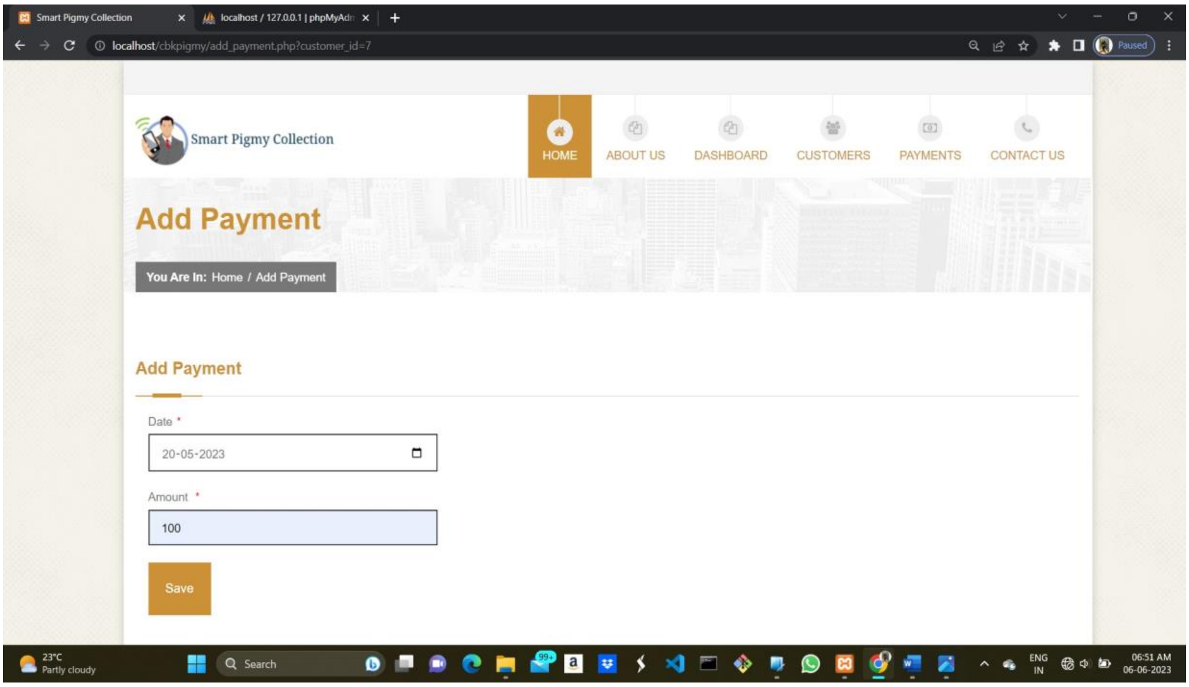
Customer Name

Search

ID	CUSTOMER NAME	MOBILE NO.	EMAIL	USER NAME	ACTIONS
7	Neha	9670270220	neha108@gmail.com	neha	Edit | Add Payment

Figure 3.1.7.7

Add Payment Page



Add Payment

You Are In: Home / Add Payment

Add Payment

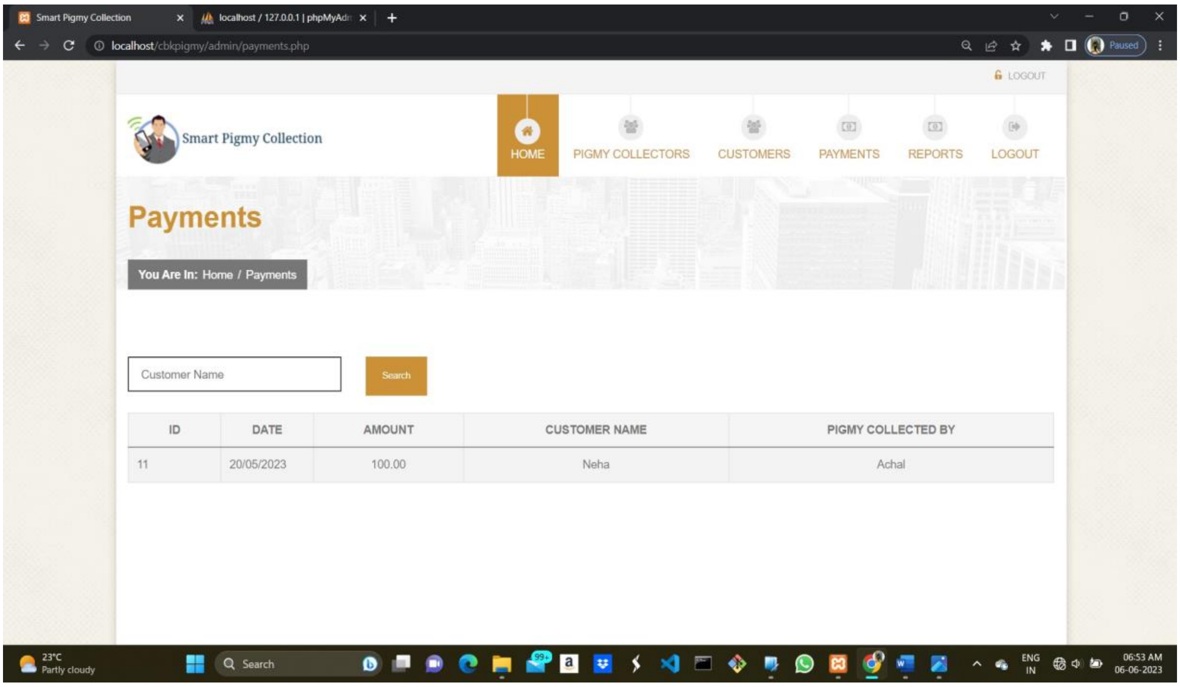
Date *
20-05-2023

Amount *
100

Save

Figure 3.1.7.8

View Payment Page



Payments

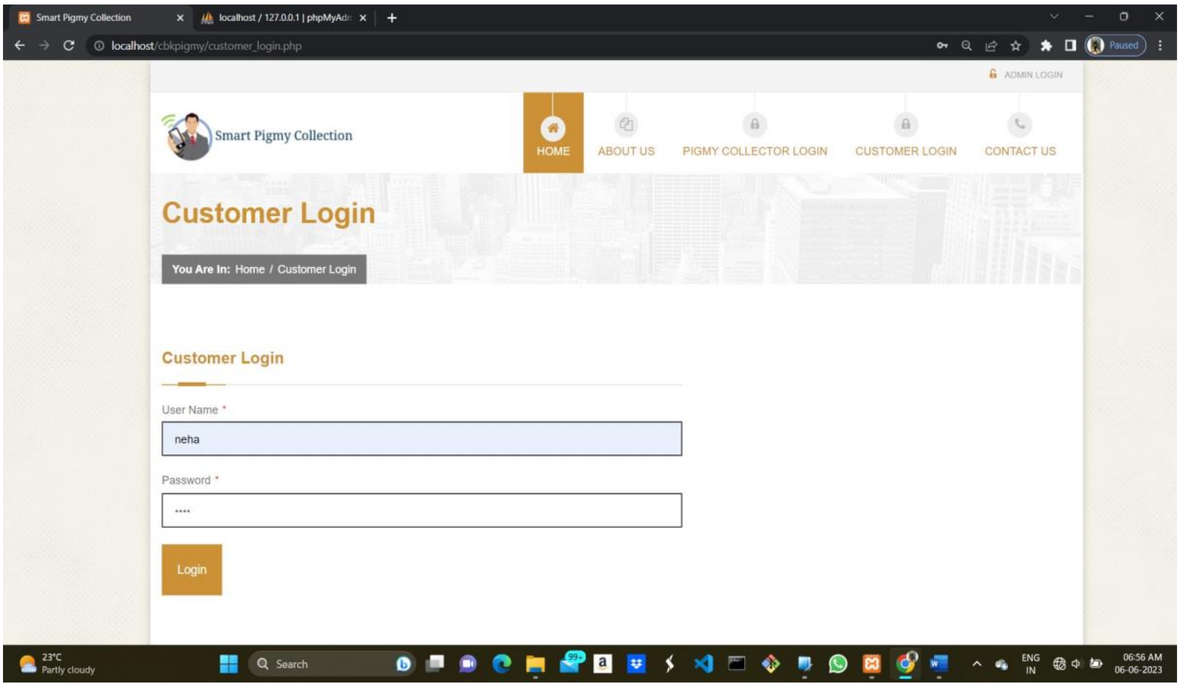
You Are In: Home / Payments

Customer Name

ID	DATE	AMOUNT	CUSTOMER NAME	PIGMY COLLECTED BY
11	20/05/2023	100.00	Neha	Achal

Figure 3.1.7.9

Customer Login Page



Customer Login

You Are In: Home / Customer Login

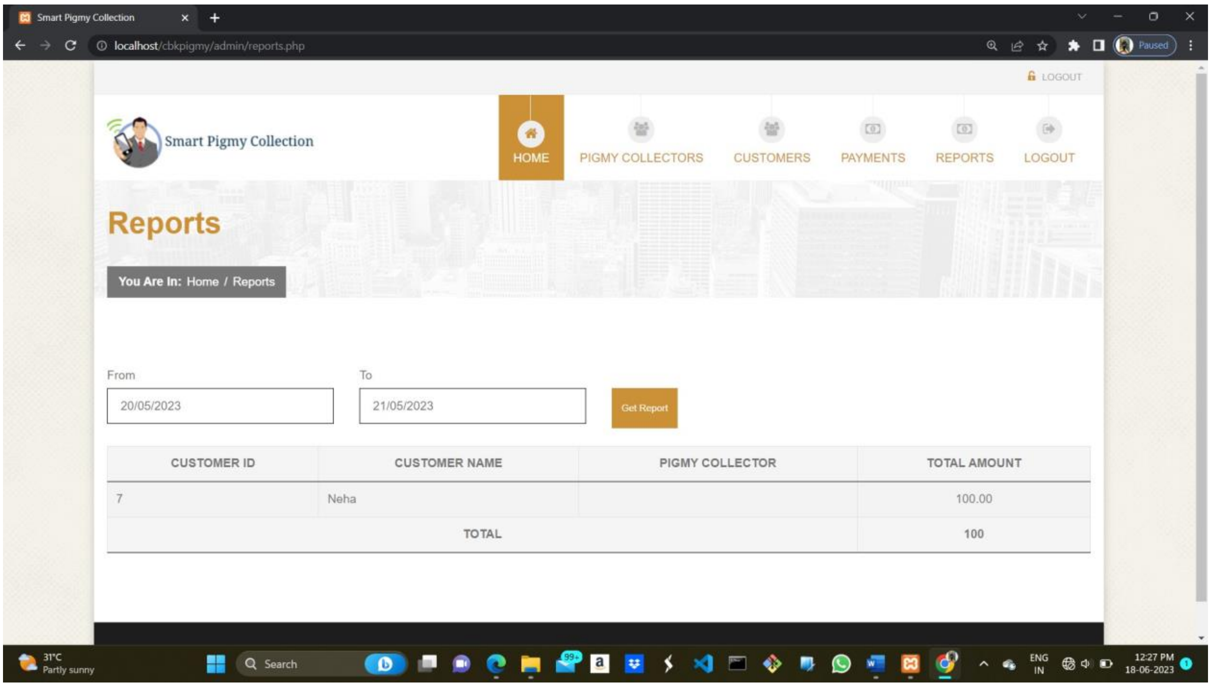
Customer Login

User Name *: neha

Password *: ••••

Figure 3.1.7.10

Report Page



Reports

You Are In: Home / Reports

From 20/05/2023

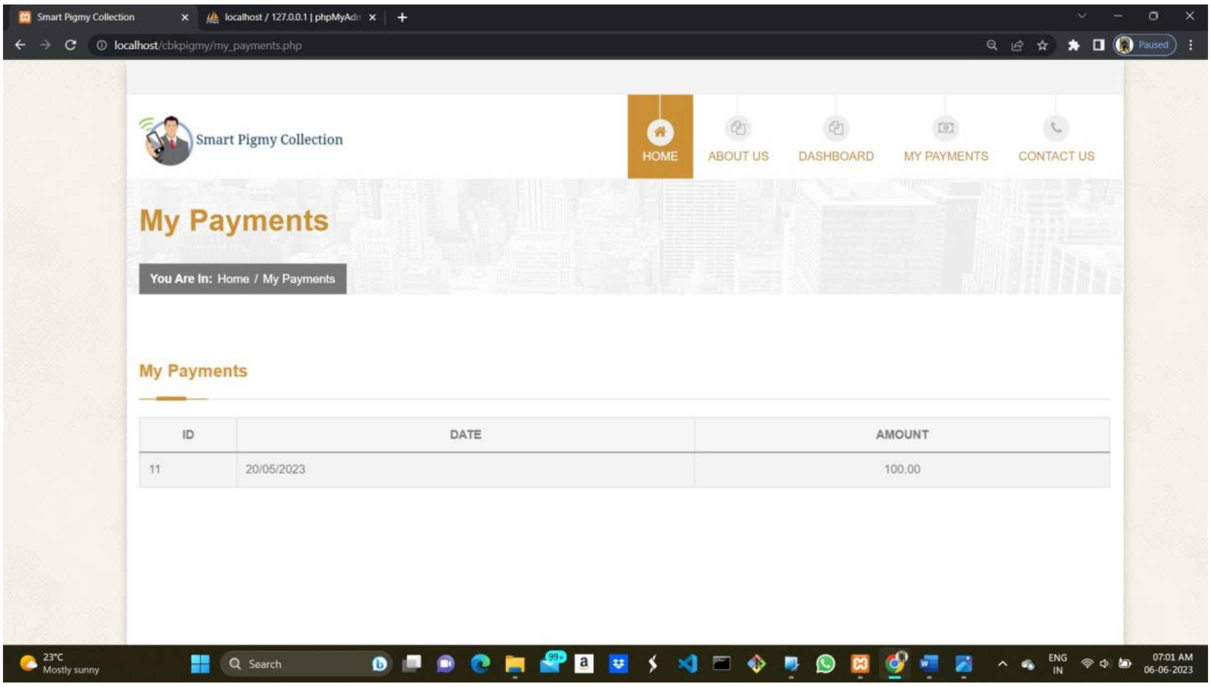
To 21/05/2023

Get Report

CUSTOMER ID	CUSTOMER NAME	PIGMY COLLECTOR	TOTAL AMOUNT
7	Neha		100.00
	TOTAL		100

Figure 3.1.7.11

My payment page



My Payments

You Are In: Home / My Payments

My Payments

ID	DATE	AMOUNT
11	20/05/2023	100.00

Figure 3.1.7.12

3.1.8 Fabrication

Admin/bank login

In admin login, first we have collected the requirements & after collecting requirements we analysed the needed requirements. Then we are designing the admin login form. After designing we are validating the form with filling the required details of each field with proper information. After we performed the database connection. Then, after completing all the steps testing will takes place.

Manage pigmy collection

In this the admin is going to manage the pigmy collection which has been collected by the pigmy collector, we have collected the requirements & analysed the requirements. Then we designed the page. After designing we have validated this page with entering some details. Then we performed the database connection. After this testing will take place.

View pigmy collection

Admin can also manage or view the pigmy collection using username and password, we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed this page. Then we have validated the form. Then we went for database connection. Then we have tested the form with comparing each step of the form.

View customer

Customer can login using their valid details. Also he can view its payment details using username and password, we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed this page. Then we have validated the form. Then we went for database connection. Then we have tested the form with comparing each step of the form.

View payment

Customer can view payment details using their valid details and also admin can view the payment history, we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed this page. Then we have validated the form. Then we went for database connection. Then we have tested the form with comparing each step of the form.

Pigmy collector login

In Pigmy collector login form, we have collected requirements, after collecting requirements we analysed the requirements for this form. Then we have designed the form. Then we have validated the form with filling some details like username & password, after form validated correctly, then we went for database connection. Then we have tested the form with comparing each step of the form.

Add Customer

In add customer, we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed this page. Then we have validated the form. Then we went for database connection. Then we have tested the form with comparing each step of the form.

Add payment

In add payment, we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed this page. Then we have validated the form. Then we went for database connection. Then we have tested the form with comparing each step of the form.

Customer login

In customer login form, we have collected requirements, after collecting requirements we analysed the requirements for this form. Then we have designed the form. Then we have validated the form with filling some details like username & password, after form validated correctly, then we went for database connection. Then we have tested the form with comparing each step of the form.

View payment

In view payment form. we have collected requirements, after collecting requirements we analysed the requirements for this form. Then we have designed the form. Then we have validated the form with filling some details like username & password, after form validated correctly, then we went for database connection. Then we have tested the form with comparing each step of the form

CHAPTER 4

4.1 Test and Validation

4.1.1 Test Plan

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

4.1.2 Software testing can be divided into two steps:

Verification: it refers to the set of tasks that ensure that the software correctly implements a specific function.

Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

4.1.3 Testing Types

Manual Testing: -

Manual Testing is a kind of software testing in which a software tester develops and executes the test cases without using any automated testing tools. The main objective of manual testing is to detect the issues, bugs, and defects of a software application.

Unit Testing: -

Unit testing is a software development process in which the smallest testable parts of an application, are called units. The main objective of unit testing is to isolate written code to test and determine if it works as intended. Unit testing is an important step in the development process. If done correctly, unit tests can detect early flaws in code which may be more difficult to find in later testing stages.

Integration Testing: -

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

White Box Testing: -

The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

Black Box Testing: -

The technique of testing in which the tester doesn’t have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

4.1.2 Test Approach

Table 4.1.2.1

Admin Login

Test Case id	Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC01	Check Admin Login with Valid Data	Enter Username Enter Password Click Login	Username=Admin Password=Admin	Admin Should login to the form.	As Expected	Pass

Table 4.1.2.2

Add Pigmy Collectors

Test Case id	Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC02	Check Add Pigmy Collector with Valid Data	Enter Name Enter Mobile no Enter Email Enter Address Enter Username Enter Password Click Save	Name=Achal Mobile no = 8296784108 Email= achalpatil009@gmail.com Address=ChandShirdwad Username=achal Password=*****	Add Pigmy Collector Should login to the form.	As Expected	Pass

Table 4.1.2.3

Pigmy Collector Login

Test Case id	Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC03	Check Pigmy Collector Login with Valid Data	Enter Username Enter Password Click Login	Username=achal Password=*****	Pigmy Collector Login Should login to the form.	As Expected	Pass

Table 4.1.2.4

View Pigmy Collector

Test Case id	**Description**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
TC04	Check View Pigmy Collector with Valid Data	Check Status	Activate Or inactivate	View Pigmy Collector Status Changed Successfully	As Expected	Pass

Table 4.1.2.5

Add Customer

Test Case id	**Description**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
TC05	Check Add Customer with Valid Data	Enter Name Enter Mobile no Enter Email Enter Address Enter Username Enter Password Click Save	Name=Neha Mobile no= 9854796581 Email= neha@gmail.com Address=Nippani Username=nehu Password=****	Add Customer Should login to the form.	As Expected	Pass

Table 4.1.2.6

Add Payment

Test Case id	Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC06	Check Add Payment with Valid Data	Enter Date Enter Amount Click Login	Date=15/05/2023 Amount=100	Customer payment should be added.	As Expected	Pass

Table 4.1.2.7

Customer Login

Test Case id	Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC07	Check Customer Login with Valid Data	Enter Username Enter Password Click Login	Username=nehu Password=****	Customer Should login to the form.	As Expected	Pass

Table 4.1.2.8

Report

Test Case id	Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC08	Check Report with Valid Data	Enter Date Click Get Report	Date=20/05/2023	Successfully Get Report	As Expected	Pass

4.1.3 Features Tested

Admin Login

Manage Pigmy Collection

View Pigmy Collection

View Customer

View Payment

Pigmy Collector Login

Add Payment

Add Customer

Customer Login

View Payment

4.1.2 Features not Tested

We have not tested the project on cloud server.

4.1.5 Findings

Admin login

If admin doesn’t provide any username in username field, we will receive a pop-up message as please enter username, and we expected the same result, hence the test is passed. If admin doesn’t provide any password in password field, we will receive a pop-up message as please enter password, and we expected the same result, hence the test is passed.

Add Pigmy Collector

If pigmy collector doesn’t provide any pigmy collector name, mobile number, email, address, username, password in the pigmy collector field, we will receive a pop-up message as please enter pigmy collector name, mobile number, email, address, username, password and we expected the same result, hence the test is passed.

Pigmy Collector Login

If pigmy collector doesn’t provide any username, password in pigmy collector login page, we will receive a pop-up message as please enter username, password and we expected the same result, hence the test is passed.

View Pigmy Collector

Pigmy collector is managing the pigmy collector list, when pigmy collector make the pigmy collector status active then it is visible to all users

Add Customer

If add customer doesn’t provide any customer name, mobile number, email, address, username, password in the add customer page, we will receive a pop-up message as please enter customer name, mobile number, email, address, username, password and we expected the same result, hence the test is passed.

Add Payment

If pigmy collector doesn’t field date in the date field, we will receive a pop-up message as please enter date, and we expected the same result, hence the test is passed. If pigmy collector doesn’t field amount in the amount field, we will receive a pop-up message as please enter amount, and we expected the same result, hence the test is passed.

Customer Login

If customer doesn't provide any username in username field, we will receive a pop-up message as please enter username, and we expected the same result, hence the test is passed. If customer doesn't provide any password in password field, we will receive a pop-up message as please enter password, and we expected the same result, hence the test is passed.

Report

If admin doesn't provide any date in date field, we will receive a pop-up message as please enter date, and we expected the same result, hence the test is passed.

4.1.6 Inference :-

Admin Login

Admin Login Plays a very vital role in our project because admin manages everything related to pigmy. Admin also Plays important role in managing pigmy collection. The customers are depended on the admin for their paid amount.

Manage Pigmy Collection

Manage pigmy collection is dependent on the admin/bank. If admin does not login to the bank the pigmy collection can not be manage.

View Pigmy Collection

Here the admin and the pigmy collector plays the important role.If the admin/bank does not login or if he not updated the daily amount to the bank we can't view the pigmy collection.

View Customer

In view customer the admin is dependent because if the admin does not login then the pigmy collector can't view the customers.Here the admin should update the customers report so that the pigmy collector can view the customers.

View Payment

Here if the bank/admin login to the bank then he can view the pigmy collection of a customer ,the pigmy collector adds the daily payment and the customer can view to it but the admin should update the payment daily.

Pigmy Collector Login

Here the pigmy collector is dependent because if the pigmy collector does not login to the bank then the collected pigmy cant be added to the bank and we cant we the collection.

Add Customer

Here the pigmy collector plays the important role because if the pigmy collector does not login to the bank then the customers cant be added so the pigmy collector must login to the bank and add the customers.

Add Payment

Here the customer adds the payment to the pigmy collector and the pigmy collector adds the payment to the bank using then valid details.

Customer Login

The customer can view his payment using his proper details so here the customer can login using there proper details and he can view the payment details.

View Payment

Here if the bank/admin login to the bank then he can view the pigmy collection of a customer ,the pigmy collector adds the daily payment and the customer can view to it but the admin should update the payment daily.

CHAPTER 5

5.1 Business aspects

5.1.1 The market and economic outlook of the project

The purpose of this project is to create “Smart Pigmy Collection” in this project pigmy collector can login using username and password after login pigmy collector can add the customer and add the daily pigmy payment in system, customer can login to this system and check the deposited payment history in bank login banks can also see all customers deposit payment details.

To implement this project we are going to use full stack development languages like HTML, CSS, JavaScript, Bootstrap, PHP, MYSQL database.

5.1.2 Features

Save Time.

Reduce Manual work.

Reduce Machine cost.

Get easy report to customer.

5.1.3 How does the product/service fit into the competitive landscape?

In the competitive landscape, pigmy collection services provided by banks play a significant role in serving a specific segment of customers with irregular or small incomes. These services cater to individuals who may not have access to traditional banking facilities or find it challenging to maintain regular savings accounts.

Pigmy collection services differentiate themselves by offering convenient and personalized doorstep banking to customers. Unlike traditional savings accounts where customers have to visit the bank branch, pigmy collection agents visit the customers' locations, making it more accessible and convenient for individuals with limited mobility or time constraints.

5.1.4 Possible capstone project clients /customers

Regional Rural Banks: Regional rural banks often offer pigmy collection services to individuals in rural areas. Collaborating with a regional rural bank could provide an opportunity to develop solutions that enhance the efficiency and effectiveness of their pigmy collection operations.

Cooperative Banks: Cooperative banks also commonly provide pigmy collection services. Working with a cooperative bank could involve designing and implementing improvements to their collection processes, digital record-keeping systems, or customer engagement strategies.

Microfinance Institutions: Microfinance institutions serve low-income individuals who may benefit from pigmy collection services. Partnering with a microfinance institution could involve developing innovative ways to enhance savings mechanisms, improve accessibility for customers, or integrate technology for better tracking and reporting.

Social Enterprises: Social enterprises with a mission to address poverty or empower marginalized communities may operate pigmy collection programs. A capstone project could involve working with these enterprises to optimize their pigmy collection processes or explore innovative solutions to scale their impact.

Government Initiatives: In some countries, governments may have programs aimed at promoting financial inclusion, especially in rural areas. Collaborating with government agencies involved in these initiatives could involve developing and implementing pigmy collection strategies aligned with their objectives.

5.2 Financial Consideration

5.2.1 Capstone project budget: 48,600/-

5.2.2 Cost capstone projections needed for either for profit/non-profit options.

Table 5.2.2.1

Title	**Cost(Rupees)**
Labor Cost	25,410
Cost of material	8,500
Net Profit	14,690
Total	48,600

5.3 Conclusion and Recommendations

5.3.1 Conclusion

As per the project we will implement the "Smart Pigmy Collection" portal we will implement all the modules like Bank login , Pigmy Collector Login, Customer login, Add payment ,Add customer, View payment history with successfully error free user friendly.

5.3.2 Future work

We can implement this project in live market.

In future we can develop in mobile application.

Program code

Index.php

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Smart Pigmy Collection</title>
<meta name="description" content="">
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
<!-- CSS StyleSheets -->
<link rel="stylesheet" href="css/font-awesome.min.css"/>
<link rel="stylesheet" href="css/animate.css"/>
<link rel="stylesheet" href="css/prettyPhoto.css"/>
<link rel="stylesheet" href="css/slick.css"/>
<link rel="stylesheet" href="rs-plugin/css/settings.css"/>
<link rel="stylesheet" href="css/style.css"/>
<link rel="stylesheet" href="css/responsive.css"/>
<!--[if lt IE 9]>
<link rel="stylesheet" href="css/ie.css">
<script type="text/javascript" src="js/html5.js"></script>
<![endif]-->
<link rel="stylesheet" id="skinCSS" href="css/skins/default.css"/>
<?php
require_once "pigmyhelper.php";
$helper = new PigmyHelper();
?>
</head>
<body style="overflow:hidden" class="bg3">

<!-- site preloader start -->
```

```
<div class="page-loader">
<div class="loader-in"></div>
</div>
<!-- site preloader end -->
<div class="pageWrapper fixedPage">
<?php
require_once "header.php";
?>
<!-- Content Start -->
<div id="contentWrapper">
<!-- Revolution slider start -->
<div class="tp-banner-container">
<div class="tp-banner" >
<ul>
<li data-slotamount="7" data-transition="zoomin" data-masterspeed="1000"
data-saveperformance="on">
<img alt="" src="images/slider/slide1.jpg" data-
lazyload="images/slider/slide1.jpg">
</li>
<li data-slotamount="7" data-transition="zoomin" data-masterspeed="1000"
data-saveperformance="on">
<img alt="" src="images/slider/slide1.jpg" data-
lazyload="images/slider/slide1.jpg">
</li>
</ul>
</div>
</div>
<!-- Revolution slider end -->

<!-- Welcome Box start -->
<div class="dark-bg" style="padding: 20px 0;">
<div class="container">
<div class="row">
<div class="cell-12">
```

```
<h2 align="center"><span class="main-color text-center">Welcome To Smart
Pigmy Collection</span></h2>
</div>

</div>
</div>
</div>
<!-- Welcome Box end -->
<div class="sectionWrapper">
<div class="container">
<div class="row">
<div class="cell-3 service-box-2 fx" data-animate="fadeInDown">
<div class="box-2-cont">
<i class="fa fa-check-square-o"></i>
<h4>Pigmy Collector Login</h4>
<a class="r-more main-color" href="#">Read More</a>
</div>
</div>
<div class="cell-3 service-box-2 fx" data-animate="fadeInDown" data-
animation-delay="200">
<div class="box-2-cont">
<i class="fa fa-check-square-o"></i>
<h4>Collect Payments</h4>
<a class="r-more main-color" href="#">Read More</a>
</div>
</div>
<div class="cell-3 service-box-2 fx" data-animate="fadeInDown" data-
animation-delay="400">
<div class="box-2-cont">
<i class="fa fa-check-square-o"></i>
<h4>Customer Login</h4>
<a class="r-more main-color" href="#">Read More</a>
</div>
</div>
```

```
<div class="cell-3 service-box-2 fx" data-animate="fadeInDown" data-
animation-delay="600">
<div class="box-2-cont">
<i class="fa fa-check-square-o"></i>
<h4>View Payments</h4>
<a class="r-more main-color" href="#">Read More</a>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Content End -->
<?php
require_once "footer.php";
?>
<div id="to-top" class="main-bg"><span class="fa fa-chevron-
up"></span></div>
</div>
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/waypoints.min.js"></script>
<script type="text/javascript" src="rs-
plugin/js/jquery.themepunch.tools.min.js"></script>
<script type="text/javascript" src="rs-
plugin/js/jquery.themepunch.revolution.min.js"></script>
<script type="text/javascript" src="js/jquery.animateNumber.min.js"></script>
<script type="text/javascript" src="js/slick.min.js"></script>
<script type="text/javascript" src="js/jquery.easypiechart.min.js"></script>
<script type="text/javascript" src="js/jquery.prettyPhoto.js"></script>
<script type="text/javascript" src="js/jquery.sharrre.min.js"></script>
<script type="text/javascript" src="js/jquery.elevateZoom-3.0.8.min.js"></script>
<script type="text/javascript" src="js/jquery.placeholder.js"></script>
<script type="text/javascript" src="js/script.js"></script>
</body>
```

```
</html>
```

Admin.php

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Smart Pigmy Collection</title>
<meta name="description" content="">

<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
<!-- CSS StyleSheets -->
<link rel="stylesheet" href="../css/font-awesome.min.css"/>
<link rel="stylesheet" href="../css/animate.css"/>
<link rel="stylesheet" href="../css/prettyPhoto.css"/>
<link rel="stylesheet" href="../css/slick.css"/>
<link rel="stylesheet" href="../rs-plugin/css/settings.css"/>
<link rel="stylesheet" href="../css/style.css"/>
<link rel="stylesheet" href="../css/responsive.css"/>
<!--[if lt IE 9]>
<link rel="stylesheet" href="css/ie.css">
<script type="text/javascript" src="../js/html5.js"></script>
<![endif]-->
<link rel="stylesheet" id="skinCSS" href="../css/skins/default.css"/>
<script type="text/javascript">
function validate_form()
{
var username = document.getElementById("username").value;
var password = document.getElementById("password").value;

if(username=='')
{
```

```
alert("Please Enter User Name.");
return false;
}
else if(password=='')
{
alert("Please Enter Password.");
return false;
}
}
</script>
<?php
require_once "adminhelper.php";
$helper = new AdminHelper();
$error = $_GET['error'];
?>
<body style="overflow:hidden" class="bg3">

<!-- site preloader start -->
<div class="page-loader">
<div class="loader-in"></div>
</div>

<!-- site preloader end -->
<div class="pageWrapper fixedPage">
<?php
require_once "header.php";
?>
<!-- Content Start -->
<div id="contentWrapper">
<div class="page-title title-1">
<div class="container">
<div class="row">
<div class="cell-12">
<h1 class="fx" data-animate="fadeInLeft">Login</h1>
```

```
<div class="breadcrumbs main-bg fx" data-animate="fadeInUp">
<span class="bold">You Are In:</span><a href="#">Home</a><span
class="line-separate">/</span><span>Login</span>
</div>
</div>
</div>
</div>
</div>
<div class="sectionWrapper">
<div class="container">
<div class="row">
<div class="cell-5 contact-form" >
<h3 class="block-head">Login</h3>
<?php
if($_GET['error'])
{
?>
<div data-animate="fadeInLeft" class="box error-box fx animated fadeInLeft"
style="">
<a href="#" class="close-box"><i class="fa fa-times"></i></a>
<h3 style="text-transform: none;">Login Failed! Please enter valid details.</h3>
</div>
<?php
}
?>
<form action="checkuser.php" id="reg_form" method="post" onsubmit="return
validate_form();">
<div class="form-input">
<label>User Name<span class="red">*</span></label><input type="text"
id="username" name="username" />
</div>
<div class="form-input">
<label>Password<span class="red">*</span></label><input type="password"
id="password" name="password" />
```

```
</div>
<div class="form-input">
<input type="submit" class="btn btn-large main-bg"
value="Login"/>&nbsp;&nbsp;
<input type="reset" class="btn btn-large" value="Reset"/>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
<!-- Content End -->
<?php
require_once "footer.php";
?>
<!-- Back to top Link -->
<div id="to-top" class="main-bg"><span class="fa fa-chevron-
up"></span></div>
</div>
<script type="text/javascript" src="../js/jquery.min.js"></script>
<script type="text/javascript" src="../js/waypoints.min.js"></script>
<script type="text/javascript" src="../rs-
plugin/js/jquery.themepunch.tools.min.js"></script>
<script type="text/javascript" src="../rs-
plugin/js/jquery.themepunch.revolution.min.js"></script>
<script type="text/javascript" src="../js/jquery.animateNumber.min.js"></script>
<script type="text/javascript" src="../js/slick.min.js"></script>
<script type="text/javascript" src="../js/jquery.easypiechart.min.js"></script>
<script type="text/javascript" src="../js/jquery.prettyPhoto.js"></script>
<script type="text/javascript" src="../js/jquery.sharrre.min.js"></script>
<script type="text/javascript" src="../js/jquery.elevateZoom-
3.0.8.min.js"></script>
<script type="text/javascript" src="../js/jquery.placeholder.js"></script>
```

```
<script type="text/javascript" src="../js/script.js"></script>
</body>
</html>
```

References

Books:-

[1] "Web Programming", by 'Chris Bates' Wiley Dream tech India, 2nd Edition.

[2] "Software Engineering", Ian Somerville, Sixth Edition, Pearson Education Ltd.

[3] "HTML Complete References" Easy steps to develop web pages.

[4] "PHP Complete Reference"

Websites:-

[1] http://en.wikipedia.org/wiki/**PHP** **for PHP.**

[2] http://www.hotscripts.com/category/**php**/ for **PHP.**

[3] http://www.**mysql**.com/click.php?e=35050 for **MYSQL.**

[4] http://www.w3schools.com **for information on HTML.**

Appendices

Chapter 1

Chapter 2

Chapter 3

List of Figures: -

List of Tables: -