**A Project report on**

**Budget and Expense Tracker System**

SUBMITTED BY

Agila Shree.A-220701015

Ajitha. B -220701019

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS) THANDALAM

CHENNAI-602105

2023 – 24

**BONAFIDE CERTIFICATE**

Certified that this project report “Budget and Expense Tracker System**”** is the bonafide work of **“AJITHA.B -220701019, AGILA SHREE.A-220701015”**who carried out the project work under my supervision

**Submitted for the Practical Examination held on**

**SIGNATURE SIGNATURE Dr.R.SABITHA Ms.S.VINOTHINI**

Professor Assistant Professor

Computer Science And Engineering Computer Science And Engineering Rajalakshmi Engineering College, Rajalakshmi Engineering College

Chennai - 602 105 Chennai - 602 105

**ABSTRACT**

The Budget Expense Tracker System is designed to streamline financial transactions, significantly reducing wait times and enhancing service efficiency. The system facilitates quick and accurate processing of expenses, which helps to avoid long queues typically associated with manual processing. In this system, the admin acts as the controller, overseeing the generation and management of transactions, while the staff users are responsible for providing services to customers.

The system is developed using a combination of web technologies to ensure a robust and user-friendly interface. PHP and MySQL are employed for the server-side processing and database management, ensuring secure and efficient data handling. The front-end of the system is built with HTML, CSS, and Bootstrap, providing a responsive and aesthetically pleasing user interface. AJAX and JavaScript are integrated to enable asynchronous operations, enhancing the system's interactivity and performance by allowing data to be fetched and processed without needing to reload the entire page. Modal dialogs are used to create a dynamic and seamless user experience for various transactional processes.

Overall, the Budget Expense Tracker System leverages modern web technologies to offer a comprehensive solution for managing financial transactions efficiently, thereby improving the service delivery process for staff and reducing wait times for customers.

**TABLE OF CONTENTS**

**Chapter 1**

**1. Introduction** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1.1 Introduction ………………………………………..………………… 1

1.2 Objective ………………………………………..…………………….1

1.3 Module ……………………………………………………...…..……..1

**Chapter 2**

**2. Survey of Technologies** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

2.1 Software description …………….………….…………………. …….2

2.2 Languages …….……………………...…………………………….….2

2.2.1 HTML ………….…………………...……………………………….2

2.2.2 CSS .………………………………….…………………………..….2

2.2.3 PHP …………………………………………………………………3

2.2.4 MySQL ………………………..…………………………...………..3

2.2.5 Bootstrap…………………………………………….……………….3

2.2.6.Ajax…………………………………...……………………………...3

2.2.7.JavaScript……………………………………..……………………...3

**Chapter 3 3. Requirements and analysis \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

3.1 Requirement Specification …..………………………….…………….. 4

3.2 Hardware and Software Requirements ….……………………………...4

3.3 Architecture Diagram ……………..………………………………..….11

3.4. ER-Diagram …………………………...…………………………..….11

**Chapter 4**

**4. Program code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

4.1 Code Details and Code Efficiency ………..………………………..…17

**Chapter 5**

**5.Conclusion** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

5.1 conclusion ……………………………………………..………………26

**Chapter 6**

**6.References** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

6.1 References……………………………………………..………………28

**Chapter 1**

**Introduction**

* 1. **Introduction**

Efficient financial management and swift transaction processing are critical components for any organization seeking to provide exemplary customer service. The Budget Expense Tracker System is designed to address these needs by reducing transaction times and preventing long queues, thereby enhancing the overall service experience. This report provides an in-depth examination of the system's development and implementation, which leverages a combination of modern web technologies to deliver a robust and user-friendly solution.

The system is structured with a clear division of roles: the admin, acting as the controller, is responsible for generating and managing transactions, while the staff users provide direct services to customers. This delineation ensures a streamlined workflow and effective oversight of financial operations.

Developed using PHP and MySQL, the system ensures secure and efficient server-side processing and database management. The front-end is built with HTML, CSS, and Bootstrap, which provide a responsive and visually appealing user interface. The incorporation of AJAX and JavaScript allows for asynchronous operations, enhancing the system's interactivity and performance by enabling real-time data processing without the need for page reloads. Modal dialogs are utilized to create a dynamic user experience, facilitating seamless transaction handling and user interaction.

This report will detail the development process, system architecture, and the technologies employed in creating the Budget Expense Tracker System. The objective is to demonstrate how the integration of these technologies results in a comprehensive solution that not only meets but exceeds the requirements for efficient financial transaction management, ultimately leading to improved service delivery and customer satisfaction.

**1.2 Objectives**

The primary objectives of the Budget Expense Tracker System are:

1. **Streamline Financial Management**: To provide a robust platform for managing financial transactions effectively,ensuring accuracy and accountability.
2. **User-Friendly Interface**: To develop an intuitive and responsive interface that enhances user experience for both admin and staff.
3. **Real-Time Data Processing**: To implement asynchronous data processing to allow real-time updates and interactions without page reloads.
4. **Secure Data Handling**: To ensure secure storage and management of financial data through reliable server-side and database technologies.

**1.3 Modules**

**1. Admin Module:**

* **Transaction Management** : Allows the admin to generate, oversee, and manage all financial transactions.
* **User Management**: Facilitates the creation, updating, and deletion of user accounts and roles.
* **Reporting and Analytics**: Provides tools for generating financial reports and analytics to monitor system performance and financial health.

**2. Database Module:**

* **Data Storage**: Securely stores all data, user information, and system logs.
* **Data Retrieval:** Efficiently retrieves data for real-time processing and reporting.

**3. Front-End Module:**

* **User Interface**: Built with HTML, CSS, and Bootstrap for a responsive and visually appealing interface.
* **Dynamic Interaction**: Uses AJAX and JavaScript to enable real-time updates and interactivity without full page reloads.

**4. Security Module:**

* **Authentication and Authorization**: Ensures that only authorized users can access certain features and data.
* **Data Encryption:** Protects sensitive financial data via encryption techniques. This achieves efficient solution for managing finances.

**Chapter 2**

**Survey of Technologies**

The development of the Budget Expense Tracker System leverages a variety of modern web technologies to achieve a robust, efficient, and user-friendly solution. This survey provides an overview of the key technologies used in the system and their respective roles.

**1.** **PHP (Hypertext Pre-processor**:

**Role**: Server-side scripting language.

**Usage**: PHP is used for backend development, handling server-side logic, processing user inputs, interacting with the database, and ensuring secure data transactions. Its ease of integration with HTML makes it an ideal choice for dynamic web applications.

**Advantages:**

* + - Open-source and widely supported.
    - Efficient handling of server-side tasks.
    - Seamless integration with various databases.

**2. MySQL:**

**Role:** Relational database management system.

**Usage:** MySQL is utilized for storing, retrieving, and managing all the data related to transactions, users, and system logs. It ensures data integrity and supports complex queries necessary for reporting and analytics.

**Advantages:**

* High performance and reliability.
* Robust security features.
* Scalability to handle growing amounts of data.

**3.HTML (Hyper Text Markup Language)**

**Role:** Standard markup language for creating web pages.

**Usage:** HTML is used to structure the content and layout of the web pages. It provides the basic framework of the system’s user interface.

**Advantages:**

* + - Easy to learn and use.
    - Universally supported by all web browsers.

**4.CSS (Cascading Style Sheets)**

**Role:** Style sheet language.

**Usage:** CSS is employed to style and layout the web pages, ensuring a visually appealing and responsive design. It separates content from design, allowing for more flexible and maintainable code.

**Advantages:**

* Enhances the aesthetic appeal of web pages.
* Improves user experience through responsive design.
* Simplifies maintenance and updates.

**5. Bootstrap**

**Role:** Front-end framework.

**Usage:** Bootstrap is used to develop responsive and mobile-first web pages. It provides a collection of pre-designed components and utilities, speeding up the development process.

**Advantages:**

* + - Facilitates rapid development.
    - Ensures consistent design across different devices.
    - Extensive documentation and community support.

**6. JavaScript**

**Role:** Programming language for web development.

**Usage:** JavaScript is used for client-side scripting to create dynamic and interactive web pages. It controls the behaviour of different elements, enabling real-time updates and validations.

**Advantages:**

* Enhances interactivity and user engagement.
* Widely supported across all web browsers.
* Extensive ecosystem with numerous libraries and frameworks.

**7. AJAX (Asynchronous JavaScript and XML)**

**Role:** Technique for creating asynchronous web applications.

**Usage:** AJAX is employed to fetch data asynchronously from the server without reloading the entire page. This ensures a smoother and faster user experience by allowing real-time updates.

**Advantages:**

* + - Improves application performance.
    - Reduces server load and bandwidth usage.
    - Enhances user experience by providing real-time feedback.

**8. Modal Dialogs**

**Role:** UI/UX component.

**Usage:** Modal dialogs are used to display content in a layered approach, without navigating away from the main page. They are particularly useful for transaction handling and user interactions.

**Advantages:**

* Provides a focused user experience.
* Simplifies complex workflows by isolating tasks

Enhances the usability of the application.

**Chapter 3**

**REQUIREMENT AND ANALYSIS**

**3.1 REQUIREMENTS SPECIFICATION**

**User Requirements:**

User requirements, a voting system can ensure a smooth, transparent, and inclusive election process that inspires trust and confidence among voters.**Top of Form**

**System Requirements:**

System requirements serve as a foundation for designing, implementing, and deploying a robust and reliable voting system that meets the needs of stakeholders and ensures the integrity of the electoral process.

**3.2 HARDWARE AND SOFTWARE REQUIREMENTS**

**Software Requirements**

* Operating System Windows 10
* Front End HTML, CSS, JavaScript
* Back End PHP, MySQL
* search

**Hardware Requirements**

● Desktop PC or a Laptop

● Printer

● Operating System – Windows 10

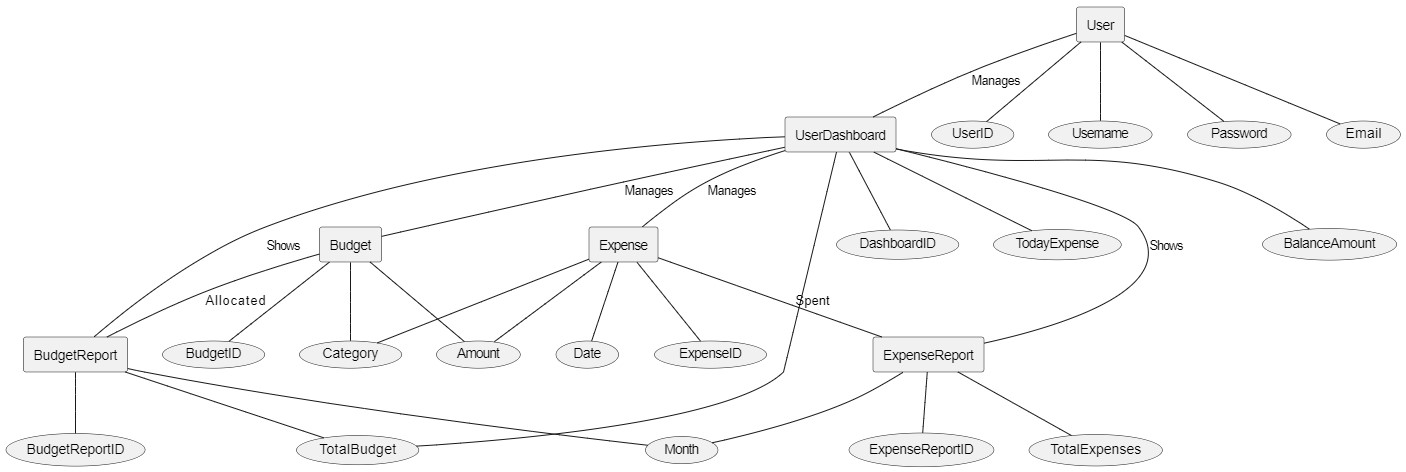
● Intel® Core TM i3-6006U CPU @ 2.00GHz

● 4.00 GB RAM

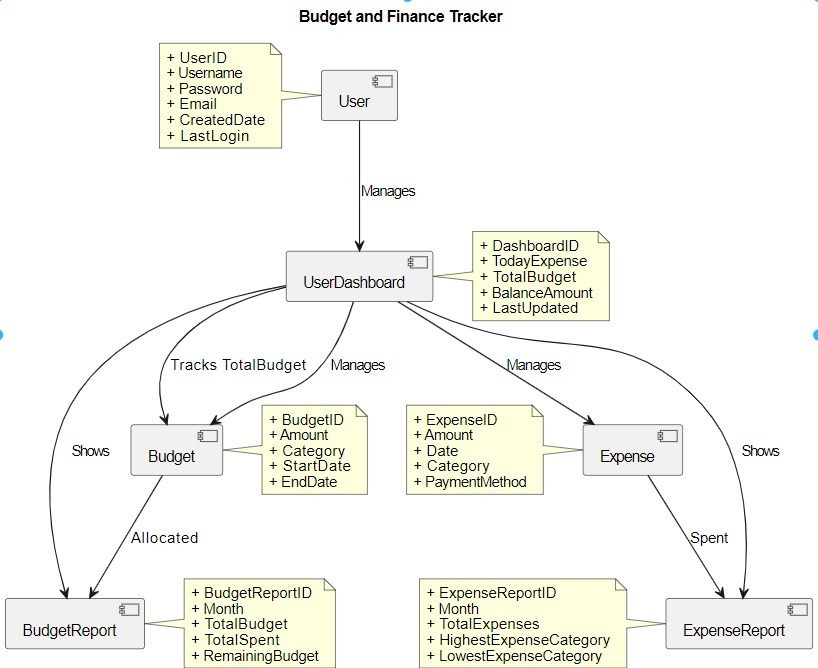
● 64-bit operating system, x64 based processor ● 1024 x 768 monitor resolution

● Keyboard and Mouse

**3.3 ER DIAGRAM**



**3.4. Architecture diagram:**



**3.5 PROGRAM**

**3.5.1.Login Page:**

<?php require\_once('../config.php') ?>

<!DOCTYPE html>

<html lang="en" class="" style="height: auto;">

<?php require\_once('inc/header.php') ?>

<body class="hold-transition login-page bg-navy">

<script>

start\_loader()

</script>

<h2 class="text-center mb-4 pb-3"><?php echo $\_settings->info('name') ?></h2>

<div class="login-box">

<!-- /.login-logo -->

<div class="card card-outline card-primary">

<div class="card-body">

<p class="login-box-msg text-dark">Sign in to start your session</p>

<form id="login-frm" action="" method="post">

<div class="input-group mb-3">

<input type="text" class="form-control" name="username" placeholder="Username" autofocus>

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-user"></span>

</div>

</div>

</div>

<div class="input-group mb-3">

<input type="password" class="form-control" name="password" placeholder="Password">

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-lock"></span>

</div>

</div>

</div>

<div class="row justify-conten-center">

<!-- /.col -->

<button type="submit" class="btn btn-primary btn-block">Sign In</button>

<!-- /.col -->

</div>

</form>

<!-- /.social-auth-links -->

<!-- <p class="mb-1">

<a href="forgot-password.html">I forgot my password</a>

</p> -->

</div>

<!-- /.card-body -->

</div>

<!-- /.card -->

</div>

<!-- /.login-box -->

<!-- jQuery -->

<script src="plugins/jquery/jquery.min.js"></script>

<!-- Bootstrap 4 -->

<script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- AdminLTE App -->

<script src="dist/js/adminlte.min.js"></script>

<script>

$(document).ready(function() {

end\_loader();

})

</script>

</body>

</html>

**3.5.2.Budgets Page:**

<?php

require\_once("../../config.php");

if(isset($\_GET['id']) && $\_GET['id'] > 0){

$qry = $conn->query("SELECT \* from `running\_balance` where id = '{$\_GET['id']}' ");

if($qry->num\_rows > 0){

foreach($qry->fetch\_assoc() as $k => $v){

$$k=stripslashes($v);

}

}

}

?>

<div class="conteiner-fluid">

<form action="" id="budget-form">

<input type="hidden" name ="id" value="<?php echo isset($id) ? $id : '' ?>">

<input type="hidden" name ="balance\_type" value="1">

<?php if(!isset($id)): ?>

<div class="form-group">

<label for="category\_id" class="control-label">Category</label>

<select name="category\_id" id="category\_id" class="custom-select select2" required>

<option value=""></option>

<?php

$qry = $conn->query("SELECT \* FROM `categories` order by category asc");

while($row= $qry->fetch\_assoc()):

?>

<option value="<?php echo $row['id'] ?>" <?php echo isset($category\_id) && $category\_id == $row['id'] ? 'selected' : '' ?> data-balance="<?php echo $row['balance'] ?>"><?php echo $row['category'] ?></option>

<?php endwhile; ?>

</select>

</div>

<?php else: ?>

<div class="form-group">

<label for="category\_id" class="control-label">Category</label>

<input type="hidden" name="category\_id" value="<?php echo $category\_id ?>">

<?php

$qry = $conn->query("SELECT \* FROM `categories` where id = '{$category\_id}'");

$cat\_res = $qry->fetch\_assoc();

?>

<p><b><?php echo $cat\_res['category'] ?></b></p>

</div>

<?php endif; ?>

<div class="form-group">

<label for="amount" class="control-label">Amount</label>

<input name="amount" id="amount" class="form-control form text-right number" value="<?php echo isset($amount) ? ($amount) : 0; ?>">

</div>

<div class="form-group">

<label for="remarks" class="control-label">Remarks</label>

<textarea name="remarks" id="" cols="30" rows="2" class="form-control form no-resize summernote"><?php echo isset($remarks) ? $remarks : ''; ?></textarea>

</div>

</form>

</div>

<script>

$(document).ready(function(){

$('.select2').select2({placeholder:"Please Select here",width:"relative"})

$('.number').on('load input change',function(){

var txt = $(this).val()

var p = (txt.match(/[.]/g) || []).length;

console.log(p)

if(txt.slice(-1) == '.' && p > 1){

$(this).val(txt.slice(0,-1))

return false;

}

if(txt.slice(-1) == '.'){

txt = txt

}else{

txt = txt.split('.')

ntxt = ((txt[0]).replace(/\D/g,''));

if(!!txt[1])

ntxt += "."+txt[1]

ntxt = ntxt > 0 ? ntxt : 0;

txt = parseFloat(ntxt).toLocaleString('en-US')

}

$(this).val(txt)

})

$('.number').trigger('change')

$('#budget-form').submit(function(e){

e.preventDefault();

var \_this = $(this)

$('.err-msg').remove();

start\_loader();

$.ajax({

url:\_base\_url\_+"classes/Master.php?f=save\_budget",

data: new FormData($(this)[0]),

cache: false,

contentType: false,

processData: false,

method: 'POST',

type: 'POST',

dataType: 'json',

error:err=>{

console.log(err)

alert\_toast("An error occured",'error');

end\_loader();

},

success:function(resp){

if(typeof resp =='object' && resp.status == 'success'){

location.reload()

}else if(resp.status == 'failed' && !!resp.msg){

var el = $('<div>')

el.addClass("alert alert-danger err-msg").text(resp.msg)

\_this.prepend(el)

el.show('slow')

$("html, body").animate({ scrollTop: \_this.closest('.card').offset().top }, "fast");

end\_loader()

}else{

alert\_toast("An error occured",'error');

end\_loader();

console.log(resp)

}

}

})

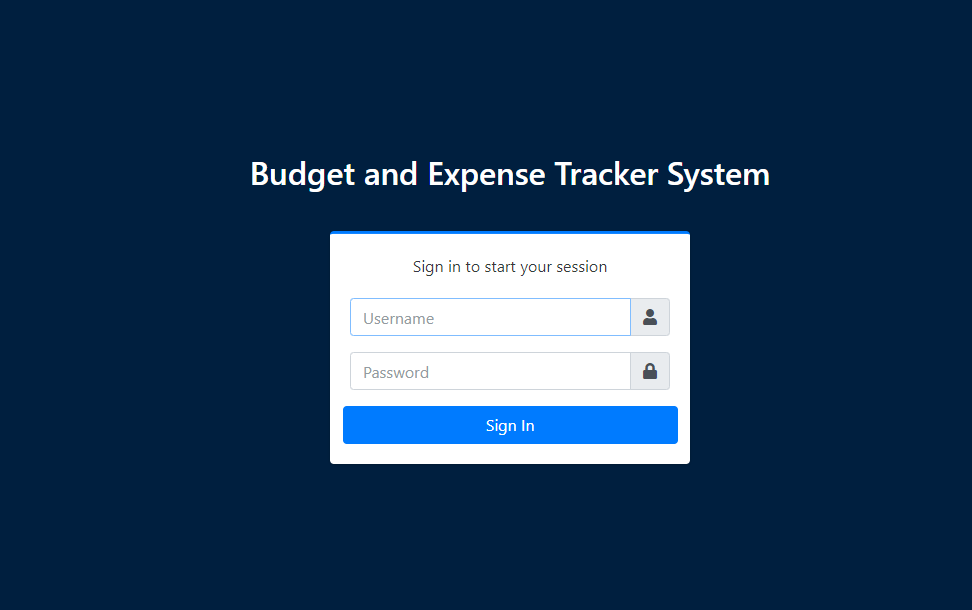
})

})

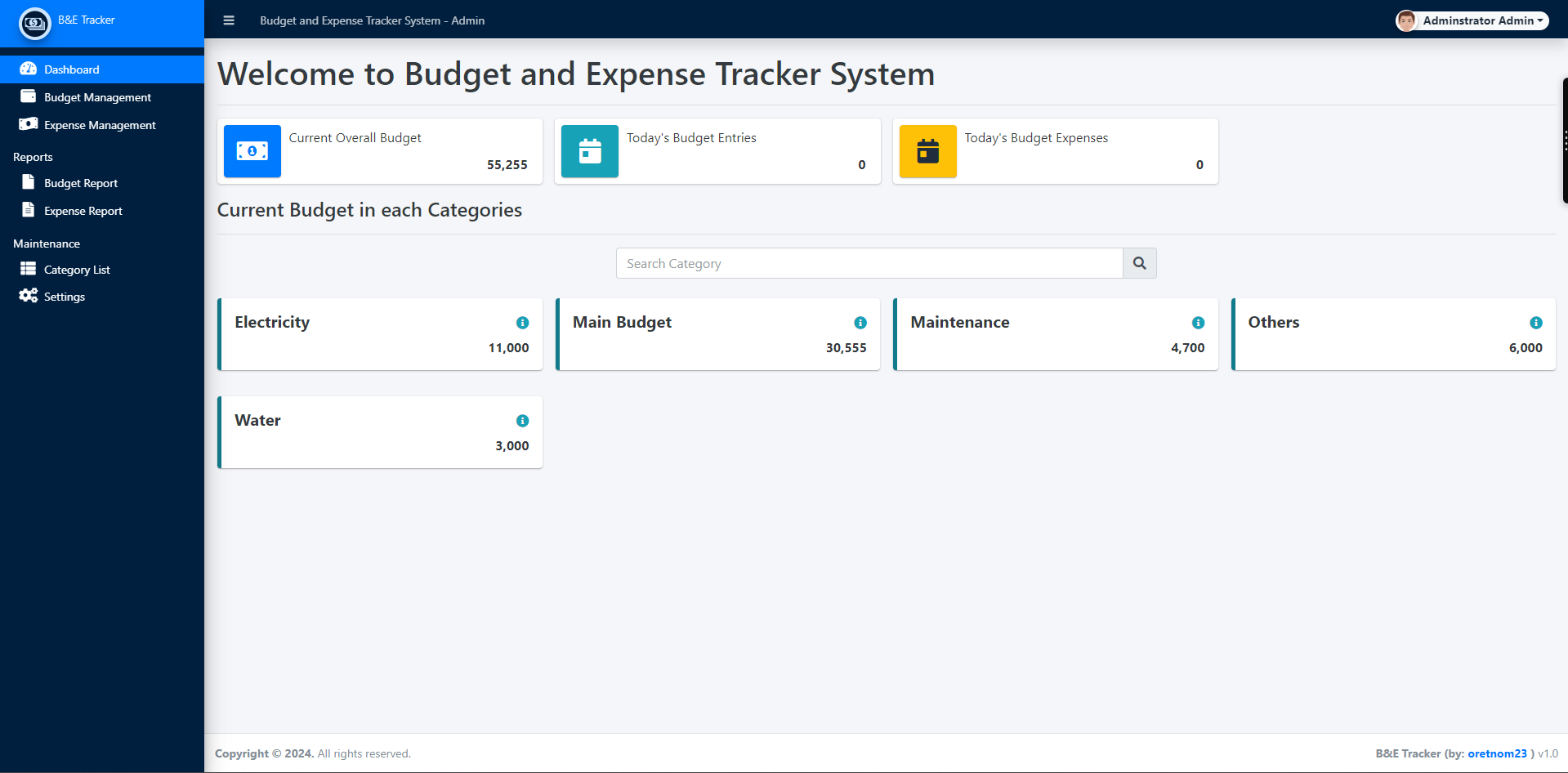
</script>

**4.RESULT AND DISCUSSION**

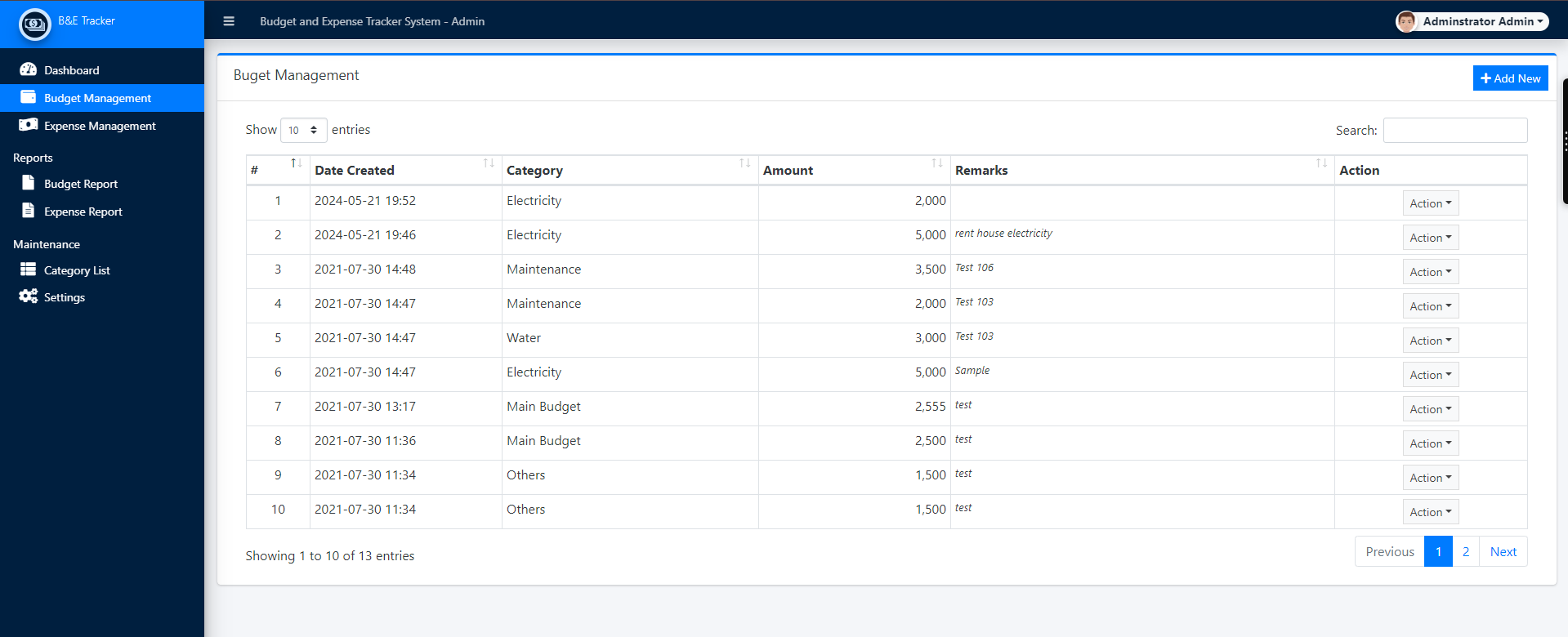
**4.1 USER DOCUMENTATION**

****

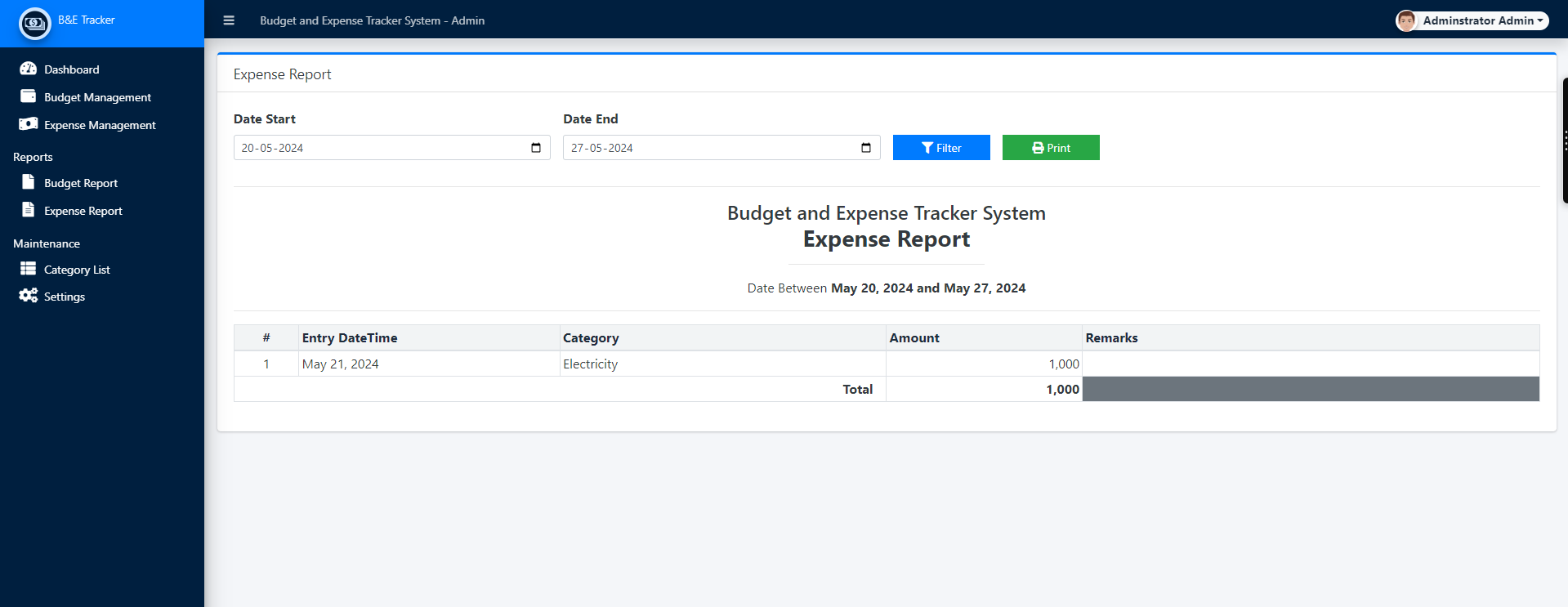
**4.1.1.Login Page**



**4.1.2.Dashboard**



**4.1.3.Budget Management Page**



**4.1.3.Budget And Expense Report Page**

**5.Conclusion:**

The development of the Budget Expense Tracker System represents a significant advancement in streamlining financial transaction processes and improving service efficiency. By integrating a range of modern web technologies, the system achieves its primary objective of reducing transaction times and preventing long queues, thereby enhancing the overall user experience for both staff and customers.

The use of PHP and MySQL provides a solid foundation for secure server-side processing and robust database management. HTML, CSS, and Bootstrap ensure a responsive and aesthetically pleasing user interface, while AJAX and JavaScript enable real-time data processing and dynamic user interactions. The implementation of modal dialogs further enhances the system's usability by offering seamless and focused transaction handling.

Through detailed functional requirements, the system supports essential operations such as user authentication, transaction management, service provisioning, real-time data processing, reporting, and user management. Additionally, the non-functional requirements ensure the system's performance, usability, security, scalability, reliability, and maintainability.

In conclusion, the Budget Expense Tracker System successfully addresses the challenges of financial transaction management by leveraging a comprehensive set of technologies. It not only meets the operational needs but also sets a new standard for efficiency and user satisfaction in financial services. The system's design and implementation provide a scalable and maintainable solution that can adapt to future needs and continue to deliver exceptional performance.

**6.REFERENCES:**

**1.Books**:

* Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson Education.
* This book provides a comprehensive understanding of database design, SQL, and database management concepts.
* Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.
* This book covers fundamental concepts of database systems, which can help in understanding the theory behind the budget tracker database design.

**2.Research Papers and Articles**:

* + Codd, E. F. (1970). *A Relational Model of Data for Large Shared Data Banks*. Communications of the ACM, 13(6), 377-387.
    - This seminal paper introduces the relational database model, which is foundational for designing relational databases like a budget tracker.
  + Date, C. J. (2009). *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media.
    - This article provides insights into writing precise SQL queries, which is essential for implementing the budget tracker database.

**3.Online Resources**:

* + W3Schools. (2023). *SQL Tutorial*. Retrieved from https://www.w3schools.com/sql/
    - This online tutorial offers practical guidance on SQL, including creating, updating, and managing databases, which is directly applicable to your budget tracker project.
  + Stack Overflow. (2023). *Database Design for Budget Tracker*. Retrieved from <https://stackoverflow.com/questions/>
    - This forum thread discusses practical aspects and challenges of designing a budget tracker database, providing community-driven insights and solutions.

**4.Documentation and Technical Reports**:

* + Oracle. (2023). *Oracle Database Documentation*. Retrieved from <https://docs.oracle.com/en/database/>
    - Oracle's official documentation is a valuable resource for understanding advanced database management features that can be leveraged in your project.
  + MySQL. (2023). *MySQL 8.0 Reference Manual*. Retrieved from <https://dev.mysql.com/doc/refman/8.0/en/>
    - This manual provides detailed information on MySQL features, which could be useful if you are using MySQL for your budget tracker.

**5.Websites and Blogs**:

* + GeeksforGeeks. (2023). *Database Management System (DBMS) Tutorial*. Retrieved from https://www.geeksforgeeks.org/dbms/
    - This site offers tutorials and explanations on various DBMS concepts, which can help in understanding the theoretical background of your project.
  + Medium. (2023). *Building a Simple Budget Tracker with SQL and Python*. Retrieved from <https://medium.com/>
    - This blog post walks through a practical example of creating a budget tracker, providing step-by-step guidance and code examples.

