Tribhuvan University

Faculty of Humanities and Social Sciences

INSTITUTE BILLING SYSTEM

A PROJECT REPORT

Submitted To

Department of Bachelor in Computer Application

Mechi Multiple Campus

In partial fulfillment of the requirements for the Bachelor in Computer Application

Submitted By

Santosh Bhandari (Roll No. 202129)

Tikaram Parajuli(Roll No. 202146)

October 2023

Under the Supervision of

Raju Poudel

Tribhuvan University

Faculty of Humanities and Social Sciences

Mechi Multiple Campus

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by TIKARAM PARAJULI AND SANTOSH BHANDARI entitled "**INSTITUTE BILLING SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Raju Poudel

SUPERVISOR

Tribhuvan University

Faculty of Humanities and Social Sciences

Mechi Multiple Campus

LETTER OF APPROVAL

This is to certify that this project prepared by SANTOSH BHANDARI AND TIKARAM PARAJULI entitled "**INSTITUTE BILLING SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In my opinion it is satisfactory in the scope and quality as a project for the required degree.

SIGNATURE of Supervisor Raju Poudel (Teaching Assistant)	**SIGNATURE OF HOD/ Coordinator**
SIGNATURE of Internal Examiner	**SIGNATURE of External Examiner**

Abstract

The Institute Billing System is a centralized Web application designed to revolutionize and optimize the billing procedures within educational institutions. This project's primary objective is to simplify and automate the billing processes, including bill generation and payment tracking, thus enhancing efficiency, precision, and financial management within the Institute.

Traditional billing systems in many educational institutions are often plagued by manual processes, leading to errors, delays, and increased administrative burdens. The Institute Billing System represents a significant advancement in addressing these challenges. It uses cutting-edge technologies to automate calculations, produce bills, and provide payment monitoring.

This project comprehensively analyzes existing billing systems, highlighting their limitations and improvement areas. It introduces the design and development of the Institute Billing System, making its key features, architecture, and functionalities. The project also encompasses a thorough feasibility study, evaluating the system's implementation's technical, economic, and operational aspects. This system will help the billing process with better services, a well-managed billing system, data integrity, data security, quick operations, and a paperless environment.

The Institute Billing System project is a pioneering initiative to streamline billing processes for educational institutions. By automating and simplifying billing operations, this system boosts efficiency and accuracy and contributes to enhanced financial management and transparency. This abstract provides a concise overview of the project's goals, methodologies, and outcomes, paving the way for educational institutes to embrace a more efficient and effective billing system.

Keywords: E-Payment, HTML5, PHP, Apache Server, Code Editor, Database, CSS, JS

Acknowledgment

We express Our sincere gratitude for the successful completion of this project, developed as part of our BCA degree program under the expert guidance and supervision of lecturer Mr. Raju Poudel.

Supervisor Mr. Raju Poudel played an instrumental role in shaping this project from its initial conception to its execution. His extensive knowledge, insightful feedback, and continuous support were invaluable throughout this journey. We have had the privilege of learning from his expertise, which has enriched our academic experience and contributed significantly to this project's quality.

We would also like to thank the program Director, Mr. Krishna Prasad Acharya, and Deputy Director, Mr. Sunil Sharma, for providing the necessary resources and environment for fostering academic growth and exploration.

Additionally, We want to acknowledge the assistance and support from our classmates and friends, who provided encouragement and valuable insights that significantly contributed to the project's development.

This project represents the culmination of efforts and support from various quarters, and we are deeply grateful to everyone who has been a part of this journey.

Santosh Bhandari

Tikaram Parajuli

Table of Contents

List of Abbreviations

DFD	Data Flow Diagram
ER	Entity Relationship
IDE	Integrated Development Environment

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

The Institute Billing System is a web application designed to revolutionize and automate billing processes within educational institutions. This innovative system tackles the challenges associated with traditional manual billing methods, which are prone to errors and delays and create a substantial administrative burden.

In an era of digital transformation, educational institutes increasingly require robust billing solutions to efficiently handle complex fee structures, generate invoices, and seamlessly track payments. The Institute Billing System rises to this challenge by providing a suite of features encompassing student information management, invoice generation, payment tracking, and robust reporting capabilities.

At its core, the system seeks to improve financial management within institutes by introducing automation and reducing reliance on error-prone manual calculations and data entry. This not only enhances efficiency but also boosts transparency in financial transactions. Moreover, the Institute Billing System delivers a user-friendly interface accessible to administrators and students, making billing information readily available and easily manageable.

This billing system allows educational institutes to streamline their financial operations, ensure accuracy, and save valuable time and resources. The system aligns with the digital demands of the modern educational landscape, guaranteeing that institutes can efficiently navigate billing complexities in the digital age.

1.2 Problem of Statement

The Manual Billing process leads to many human errors like incorrect entries and inaccuracies in fee calculations and invoices. Manual billing makes it time-consuming to track the payment details and keep the correct financial statements. This will automatically increase the burden on the administration worker. [1]

In the traditional billing process, the invoices are generated manually. Whenever the invoices are generated, the students must mention the information about them and the courses they have currently taken. Whenever the student wants to update their billing information, It will be complex in the manual billing process.

The Problem of the statement are as follows:

- Invoices are generated manually.
- Complex and Time Consuming to Track the Payment Information.

1.3 Objectives

The Objectives are as follows:

- To Generate Computerized Invoices for any Institute.
- To keep a record of Payments, Invoices, and Student Information

1.4 Scope and Limitation

The Institute Billing System will allow the admin to add student information and generate the invoices according to the information.

1. This Billing System Maintains the Student Information Systematically.
2. The admin can manage the course, admin, and student information.
3. The system generates the invoices based on the provided billing information.
4. It holds the billed information and billed date of concerned students.

The Limitations of the Institute Billing System are as follows:

1. Requires the devices that can host the PHP files.
2. Requires the devices that have a database.
3. The data are not encrypted. So, any vulnerabilities in the system could compromise sensitive student and billing information.

1.5 Report Organization

The Report of Institute Billing System is organized as follows:

Chapter 1: Introduction

This chapter provides an overview of the project, including a clear problem statement. It outlines the project's objectives, scope, and limitations, setting the stage for the report.

Chapter 2: Background Study and Literature Review

In this chapter, we dive into the project's background, comprehensively understanding its functions and components. A literature review is conducted to explore previous research related to the project, including an analysis of similar systems for evaluation.

Chapter 3: System Analysis and Design

It focuses on the systematic analysis and design of the project, illustrated with charts and figures. We delve into functional requirements using use cases and other methodologies. Additionally, this chapter presents the database schema, interface design, and DFD.

Chapter 4: Implementation and Testing

This chapter details the tools and techniques utilized for project implementation. We also cover creating test cases to assess the system as individual units and as a whole.

Chapter 5: Conclusions and Future Recommendations

It provides information about the experiences and insights gained throughout the project, offering recommendations for future projects. It concludes the report by summarizing key findings and takeaways.

Chapter 6: Appendices

Chapter 6 contains supplementary materials, including screenshots and source codes for various functions and pages of the project.

Chapter 7: References

The last chapter references the sources from which content was drawn and used within this report.

Chapter 2: Background Study and Literature Review

2.1 Background Study

The Institute Billing System project emerges in response to the pressing need for modernization within educational institutes. Traditional manual billing processes have long been a staple in the institutions but come with inherent challenges. These processes are susceptible to errors, time-consuming, and often lack transparency. The Institute Billing System aims to resolve these issues by introducing a digital solution that automates billing procedures. In an era where efficiency and accuracy are paramount, this system allows educational institutes to streamline their financial operations, enhance transparency, and provide a user-friendly platform for administrators and users to access billing information seamlessly.

As the digital age continues to reshape industries, educational institutes are no exception. The demand for robust billing systems that manage complex student information, generate accurate invoices, and track payments has never been higher. The Institute Billing System addresses this demand, offering a suite of features encompassing student information management, invoice generation, and payment tracking. Through its automation capabilities, the system reduces reliance on manual calculations and data entry, effectively minimizing errors and saving valuable time for administrative staff. In embracing the Institute Billing System, educational institutes align themselves with the digital evolution, ensuring that their billing processes are efficient and capable of meeting the evolving expectations of students, parents, and staff in today's digital landscape.

2.2 Literature Review

The Smart Billing System represents a software solution to streamline educational institutions' intricate invoice generation process. It acts as a comprehensive tool, managing various facets of billing, ranging from complicated fee calculations to the seamless generation of invoices. One of its notable strengths lies in its user-friendly interface, which simplifies navigation and accessibility. Additionally, the system incorporates basic reporting features, empowering administrators with essential financial insights. Despite these merits, the system does face certain limitations, such as challenges in handling large volumes of billing data and the reliance on manual calculations, which can introduce room for errors and inefficiencies. [2]

MIS, or Management Information System, is another software solution that handles billing processes. It stands out for its use of modern technologies and a user-friendly interface, making the billing process smoother and more efficient. Unlike some standalone systems, MIS is integrated with other systems, which can have advantages and drawbacks. This integration can enhance overall efficiency but might lead to complexities and dependencies on other systems. [3]

The Supermarket Billing System provides an efficient user interface for fast processing and long-term error-free use. These systems excel in accurate bill calculation, record-keeping, and report generation. Additionally, modern billing systems now incorporate customer relationship management, offering incentives and insights to loyal customers. The process involves product data entry, tax and commission setting by administrators, customer product selection, barcode verification, bill calculation, payment, and product delivery. This system can also be adapted to Lending and Cooperatives, simplifying billing processes and account management with user-specific limitations. [4]

The Invoice Management System web-based application is designed to efficiently handle vendor or supplier documents, particularly invoices and critical source documents in accounting. This project offers automatic invoicing, administrative controls, email invoicing, a supplier login area, customizable web pages, ticket support, and more, simplifying invoice management for timely vendor payments. With a robust backend system consisting of advanced modules, this system enhances document management, making it a valuable tool for businesses seeking efficient and flexible invoicing processes. [5]

Chapter 3: System Analysis and Design

3.1 System Analysis

3.1.1 Requirement Analysis

The requirements of the system are determined as given below.

i. Functional Requirements

Figure 3.1.1: Use Case Diagram of Institute Billing System

Login: Admin can log in to the system and get access to their privileges in the system.

Logout: Admin can log out from the system.

Add Student: Admin can add New Student records.

Update Student: Admin can update existing student information.

Delete Student: The admin can remove the student information from the database.

Add Bill: Admin can generate the bill by providing bill information.

Update Bill: Admin can update existing bill records of concerned students.

Delete Bill: Admin can remove the bill information from the database.

View Bill: Admin and Student can view the generated bill.s

Add Admin: Admin can create a new Admin.

Update Admin: Admin can Update the records of existing admin.

Delete Admin: Admin can remove the admin from the system.

Add Course: Admin can create a New Course.

Update Course: Admin can update the existing course information.

Delete Course: Admin can remove the course from the database.

ii. **Non Functional Requirements**

Non-functional requirements define system attributes such as security, reliability, performance, maintainability, scalability, and usability. The non-functional requirements of the Institute Billing System are:

- **Reliability:** This System is easy to use and doesn't have high system requirements. So, the system is reliable.
- **Compatibility:** The System should work seamlessly across different web browsers and devices to provide a consistent experience.
- **Security:** The System will be secure, and the User or admin can only access the system information according to their preimage level.
- **Usability:** The system user interface is intuitive and user-friendly, ensuring users can navigate it easily.
- **Maintainability:** The System has fewer hardware requirements and can be maintained within an affordable budget in case of technical errors.
- **Response Time:** The System consistently provides timely responses to user/admin actions to keep their experience smooth.

3.1.2 Feasibility Analysis

Before proceeding with the development of the system, a feasibility study is conducted to assess the viability and potential success of the project. This study evaluates various aspects, including technical feasibility, economic feasibility, and operational feasibility, to determine if the project is worth pursuing. [6]

i. Technical

Technical feasibility assesses whether the development and implementation of the proposed system are achievable using the available technology and resources. It involves evaluating the compatibility of the software solution with the existing infrastructure, hardware, and software systems.

Table 3.1.1 : Technical Feasibility

Parameters	Remarks
Hardware Resources	Available
Technical Expertise	Yes
Programming Language	Free
Software Tools	Open Source / Free

The technical feasibility assessment indicates the necessary technical resources and infrastructure are available.

ii. Operational

Operational feasibility assesses whether the proposed system can be smoothly integrated into the existing operations and processes. It evaluates the system's compatibility with workflow and users' acceptance.

Table 3.1.2: Operational Feasibility

Parameters	Remarks
User Acceptance	Yes
Ease of Use	Yes
Training Requirement	Yes
Operational Cost	Affordable

The project team estimated that the remaining and running system after the deployment would not be a problem, indicating that the project is operationally feasible.

iii. Economic

Economic feasibility assesses the financial viability and benefits of developing and implementing the proposed system. It involves analyzing the costs associated with the project and comparing them with the potential benefits and returns on investment.

Table 3.1.3 : Economic Feasibility

Parameters	Remarks
Development Tools	Open Source / Free
Servers	Free
IDE	Open Source / Free
Programming Language	Free

The economic feasibility study demonstrates that the proposed system can be developed cost-effectively.

iv. Schedule

In project feasibility analysis, schedule feasibility is a critical dimension to evaluate. It assesses whether the proposed project can be executed within the stipulated time frame, considering various factors that might impact project schedules.

Table 3.1.4: Schedule Feasibility

Tasks	Start	Finish	Days
Requirement Collection	7/9/2023	7/12/2023	4
Analysis	7/13/2023	7/18/2023	6
System Design	7/19/2023	7/28/2023	10
Coding	7/29/2023	9/1/2023	35
Testing	8/18/2023	9/2/2023	16
Deployment & Maintenance	9/3/2023	9/10/2023	8
Documentation	7/21/2023	9/3/2023	45

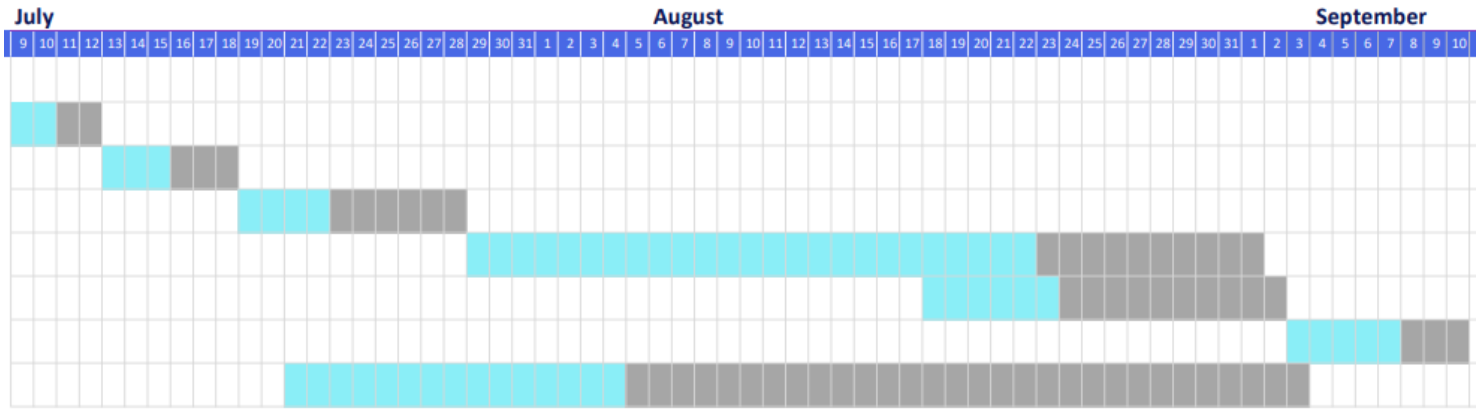

Figure 3.1.2: Gantt Chart of Institute Billing System

3.1.3 Data Modeling (ER-Diagram)

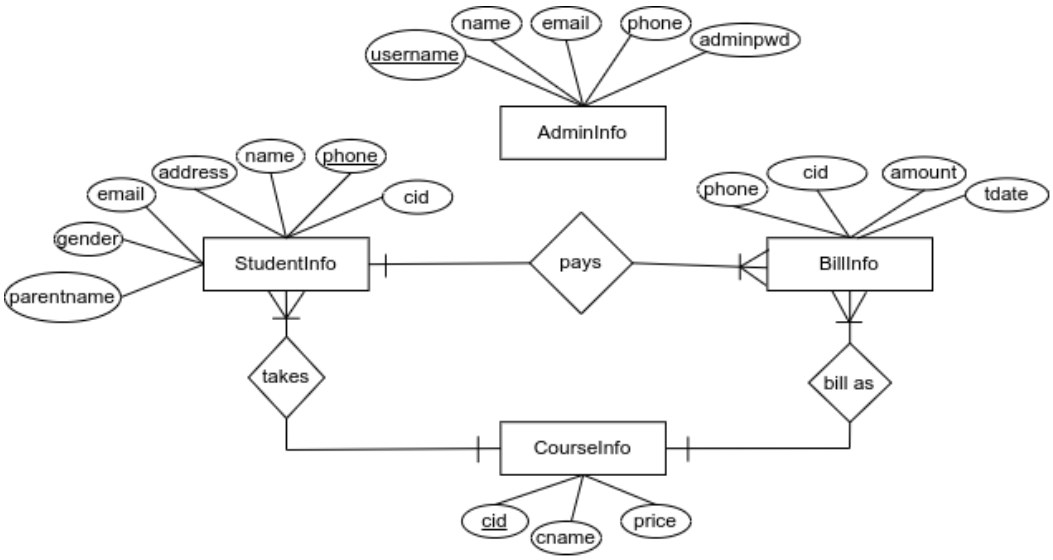

Figure 3.1.3 : ER Diagram of Institute Billing System

3.1.4 Process Modeling(DFD)

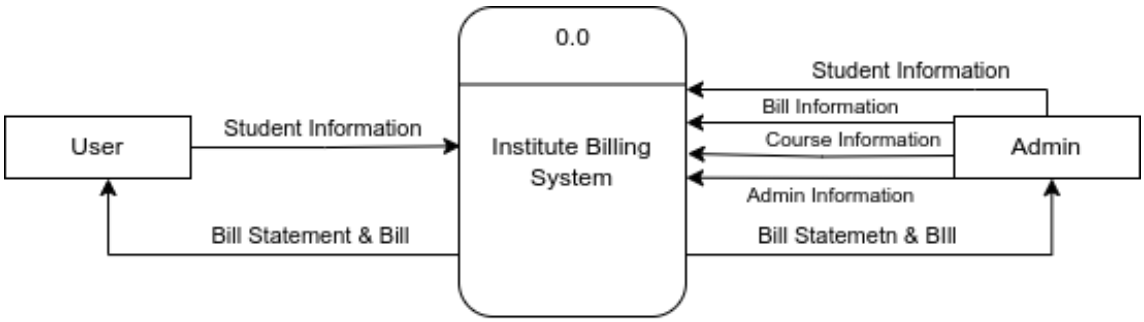

Figure 3.1.4: Context Diagram of Institute Billing System

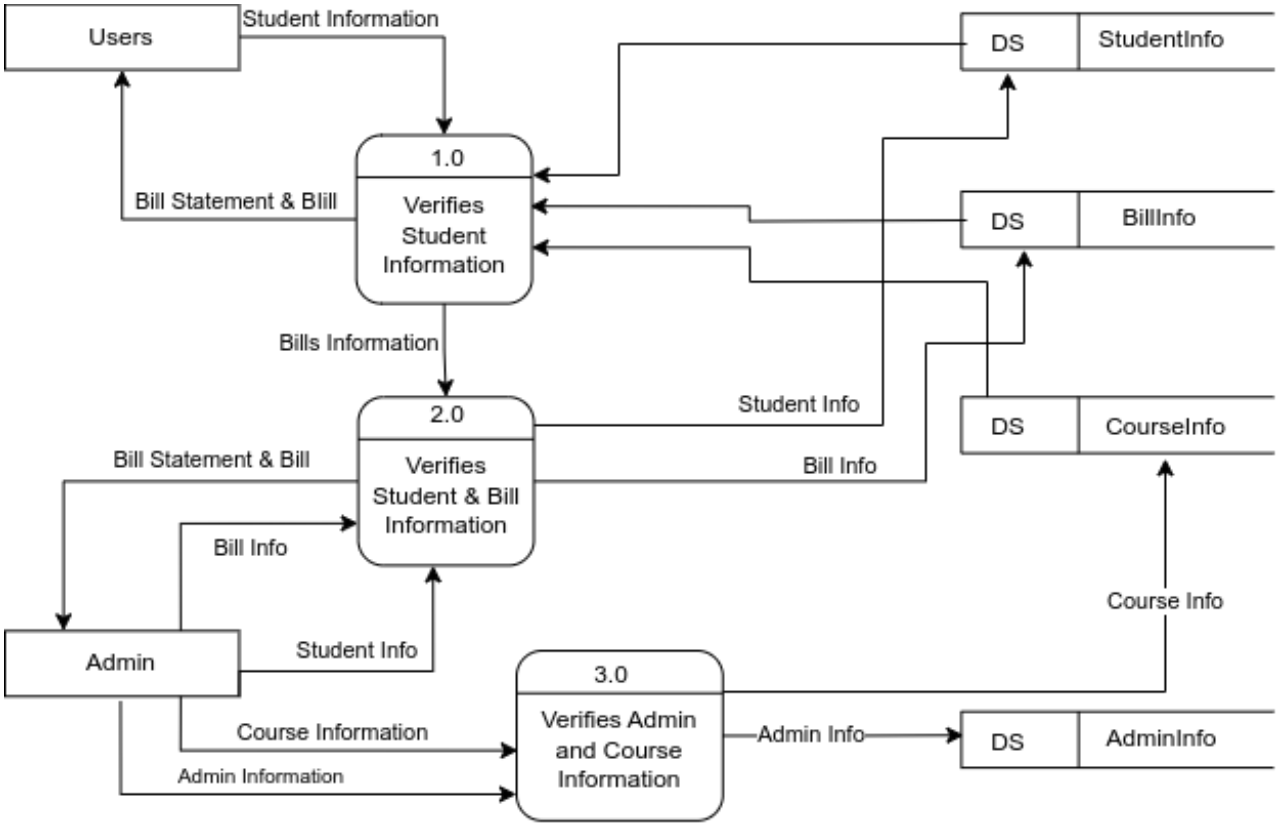

Figure 3.1.5: Level 1 DFD of Institute Billing System

3.2 System Design

3.2.1 Archietctural Design

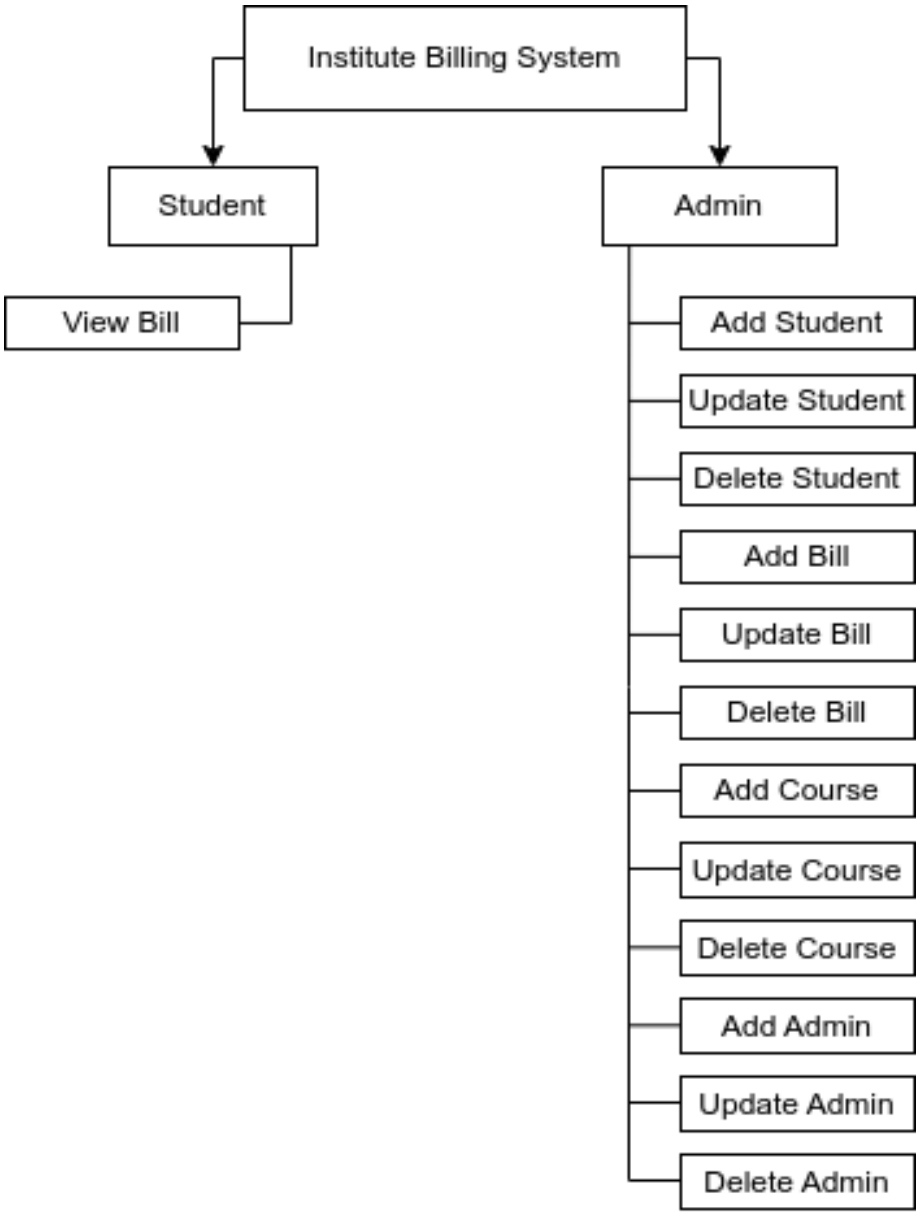

Figure 3.2.1: Architectural Design of Institute Billing System

3.2.2 Database Schema Design

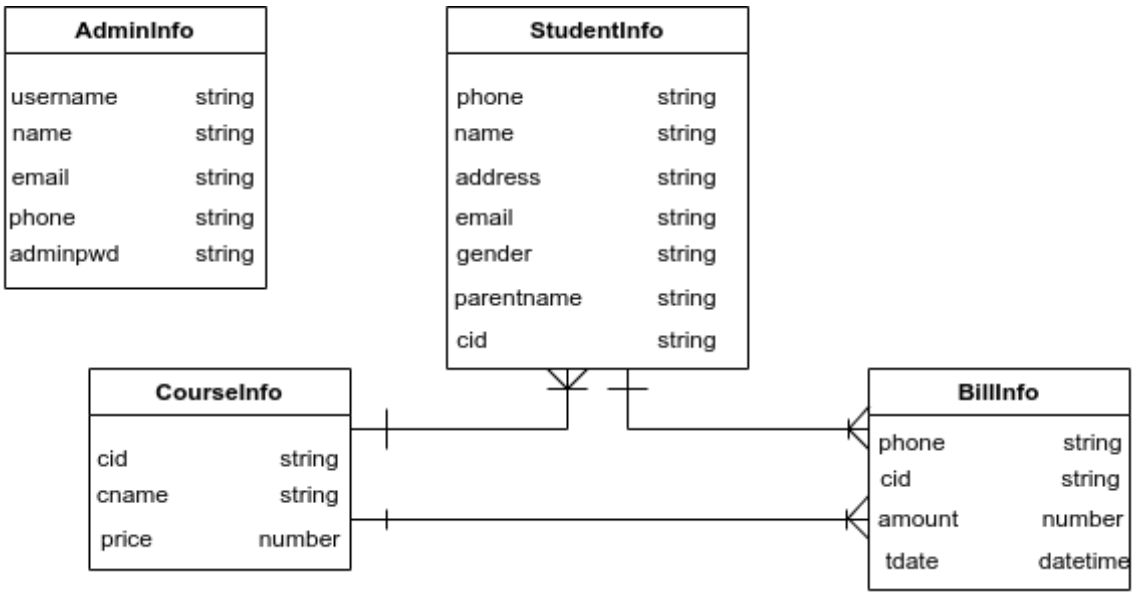

Figure 3.2.2 : Database Schema of Institute Billing System

3.2.3 Interface Design

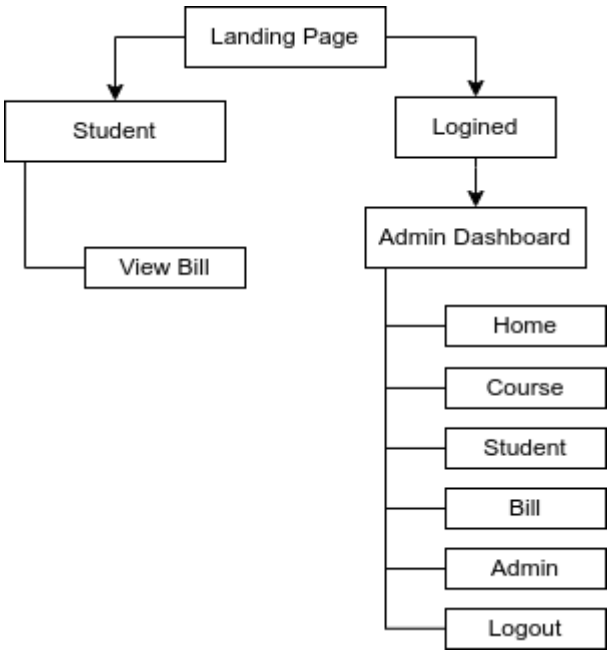

Figure 3.2.3 : Interface Design of Institute Billing System

Chapter 4: Implementation and Testing

4.1 Implementation

4.1.1 Tools Used

The following tools are used to accomplish this project:

Front End Tools

- **HTML 5**

HTML is the backbone of web pages, providing the structure and content. It uses tags to define elements such as headings, paragraphs, links, and images, allowing browsers to render web content correctly.

- **CSS 3**

CSS complements HTML by handling presentation and layout. With CSS, developers can control the visual aspects of web pages, including colors, fonts, spacing, and responsiveness. It enables the creation of visually appealing and consistent designs across different devices.

- **JS**

JavaScript is a dynamic scripting language that adds interactivity to web pages. It enables features like form validation, animations, and real-time updates without requiring a page reload. JavaScript is essential for creating engaging and user-friendly web applications.

Back End Tools

- **PHP**

This system uses PHP as a backend tool to develop dynamic web pages. It is a popular server-side scripting language commonly used for web development. It empowers developers to create dynamic web pages, process forms, interact with databases, and build robust web applications. PHP is known for its flexibility, ease of integration with various databases, and extensive community support, making it a cornerstone of web development for building dynamic and data-driven websites.

Database

- **MySQL**

MySQL stores all the required Student, Bills, Users, and Admin information in this system. It is critical in storing, managing, and retrieving structured data. It is compatible with various languages and offers advanced features.

Server

- **Apache Server**

In the Institute Billing System, Apache Server hosts and runs PHP files. It is a widely used open-source web server software.

Code Editor

- **Visual Studio Code**

VS Code is used to write code for this system. It is a free and lightweight source code editor supporting multiple programming languages.

Browser

- **Mozilla Firefox**

Mozilla Firefox is used to run the project. It is an open-source web browser that provides built-in tools for developers.

4.1.2 Implementation Details of Modules

According to the Needs, this system is divided into two main functional modules, i.e., the User module and the admin module.

Student Module: This module is designed for Students to view the bill and bill statement.

Admin Module: The admin module holds additional capabilities than the student module. It can create, delete, and update admin, bill, and course information. It also has the features to add, delete, and update student records according to the need.

4.2 Testing

Testing is the process of executing a program to find an error. A good test case has a high probability of finding an undiscovered error. The primary objective for test case design is to derive a set of tests with the highest livelihood for uncovering defects in software. To meet this objective, two different categories of test case design are used.

4.2.1 Test Cases for Unit Testing

Table 4.2.1: Test Case for Bill Statement

S.N.	Input Student Information	Expected Output	Actual Output	Result
1.	Santosh bhanari 9824988945	Show the bill statement	Showed the Billed Statement with student information.	Pass
2.	Kiran dahal 9816000000	Show Message No record found.	Showed No Record Fount Message.	Pass

Table 4.2.2: Test Case for Login

S.N.	Input Login Data	Expected Output	Actual Output	Result
1.	admin admin	Login to admin dashboard	Logged In and got access to admin dashboard.	Pass
2.	root root	Show Error message Invalid username or password.	Invalid username or password message shown.	Pass

Table 4.2.3: Test Case for Creating Admin

S.N.	Input Admin Data	Expected Output	Actual Output	Result
1.	Name: admin Email: admin@admin.com Number: 9824000000 Username: admin Password: admin123	Show Error Message Selected Admin Username Already Added	Selected Admin Username Already Added message shown.	Pass

2.	Name: Santosh Bhandari Email: Santosh@mail.com Number: 9824988945 Username:santosh Password: santosh123	Show Success message Admin Added Successfully!!	Shown Admin Added Successfully!!.	Pass
3.	Name: Ayush Email: ayush@gmail.com Number: 9824123456 Username: Password: ayush123	Failed to Submit the Form	Please fill out this field message shown.	pass

Table 4.2.4 : Test Case for Adding Course

S.N.	Input Course Data	Expected Output	Actual Output	Result
1.	Course ID: C101 Name: PHP Programming Price: 10000	Show Success Message Course Added Successfully	Shown Course Added Successfully	Pass
2.	Course ID: C101 Name: PHP Programming Price: 15000	Error Message Course Already Added	Course Already Added Message Shown.	Pass
3.	Course ID: C103 Name: Java Programming Price:	Failed to Submit the Form	Please fill out this field message shown.	pass

Table 4.2.5: Test Case for Adding Student Information

S.N.	Input Course Data	Expected Output	Actual Output	Result
1.	Name: Santosh Bhandari Address: Kanakai-07 Mobile: 9824988945 Email: santosh@gmail.com Gender: Male Parent Name: Hari Bhandari Course: Java Programming	Show Success Message Student Record Added Successfully	Student Record Added Successfully message Shown	Pass

2.	Name: Santosh Bhandari Address: Kanakai-07 Mobile: 9824988945 Email: santosh@gmail.com Gender: Male Parent Name: Hari Bhandari Course: Java Programming	Error Message Student Record Already Added.	Student Record Already Added error message shown.	Pass
3.	Name: Kiran Dahal Address: Kanakai-07 Mobile: Email: Gender: Male Parent Name: Pream Dahal Course: PHP Programming	Failed to Submit the Form	Please fill out this field message shown.	pass

Table 4.2.6: Test Case for Adding Bill Information

S.N.	Input Course Data	Expected Output	Actual Output	Result
1.	Name: Santosh Bhandari Total Fee: 15000 Due Amount: 10000 Amount: 5000	Redirect to Bill Page	Redirect and Generate the Bill	Pass
2.	Name: Santosh Bhandari Total Fee: 15000 Due Amount: 5000 Amount: 15000	Disable Submit and Show Error Message	Error Message Shown and Submit Button Disabled.	Pass
3.	Name: Santosh Bhandari Total Fee: 15000 Due Amount: 5000 Amount:	Failed to Submit the Form	Please fill out this field message shown.	pass

4.2.2 Test Cases for System Testing

Table 4.2.7: Test Cases For System Testing

S.N.	Input Data	Expected Output	Actual Output	Result
1.	Enter URL	Open Application Home Page	Applicaton Home Page Opended	Pass
2.	Santosh Bhandari 9824988945	Show the bill statement	Showed the Billed Statement with student information.	Pass
3.	Enter Login Url	Open Application Login Page	Applicaton Login Page Opended	Pass
4.	admin admin	Login with admin privileges.	Logged in and got access to admin privileges.	Pass
5.	Add Course	Add New Course	New Course Added.	Pass
6.	Update Course	Update details of existing course.	The Price and Course Name is updated	Pass
7.	Delete Course	Delete Existing Course	Existing Course Deleted.	Pass
8.	Add Student	Add New Student Information	New Student Information added.	Pass
9.	Delete Student	Delete Existing Student Information	Existing Student Information Deleted.	Pass
10.	Update Student	Update Informaiton of Existing Student.	The Name, Address, Email, Gender and Parent Name is Updated	Pass
11.	Add Bill	Create New Bill	New Bill Created.	Pass
12.	Delete Bill	Delete Bill Details	Bill Details Deleted.	Pass
13.	Update Bill	Update amount of the bill.	The amount of the bill is updated.	Pass
14.	Add Admin	Create New Admin	New Admin Created	Pass
15.	Update Admin	Update the Exisitng record of Admin.	Name, Email, Number and Passoword is Updated.	Pass
16.	Delete Admin	Delete Existing Admin Record	Existing Admin Record Deleted.	Pass

Chapter 5: Conclusion and Future Recommendations

5.1 Lesson Learnt / Outcome

The primary objective of this project was to create a working system that would help in the billing process of the Institute and track the record of the payments made by the student, as well as the features of removing or modifying existing student information. The most important lesson learned was managing time according to the system components' complexity, i.e., knowing which components to prioritize. A software developer must always research in their daily coding life, so it is important to explore things efficiently. A software developer must choose the right and easy option.

5.2 Conclusion

The Institute Billing System was created to modernize the billing process, replacing traditional methods. The key goals included the generation of computerized invoices, efficient payment tracking, and maintaining student records. With the successful development of the system, it has achieved all these predefined objectives. It now generates invoices and effectively tracks payment information while maintaining comprehensive student records.

5.3 Future Recommendations

For the improvement of the system, the following things can be done:

i. **Online Payment Gateway:** Implement an online payment gateway for students and parents to make directly through the system, increasing convenience and reducing manual payment handling.
ii. **Automated Notifications:** Integrate automated notification features to alert students and parents about due payments, dates, and payment confirmations via email or SMS.
iii. **Data Analytics:** Enhance the system with analytics capabilities to provide insights into payment trends, outstanding balances, and revenue projections for better financial planning.

These recommendations aim to enhance the functionality and user experience of the system while adapting to the evolving needs of the institutions.

Appendices

Screen Shots

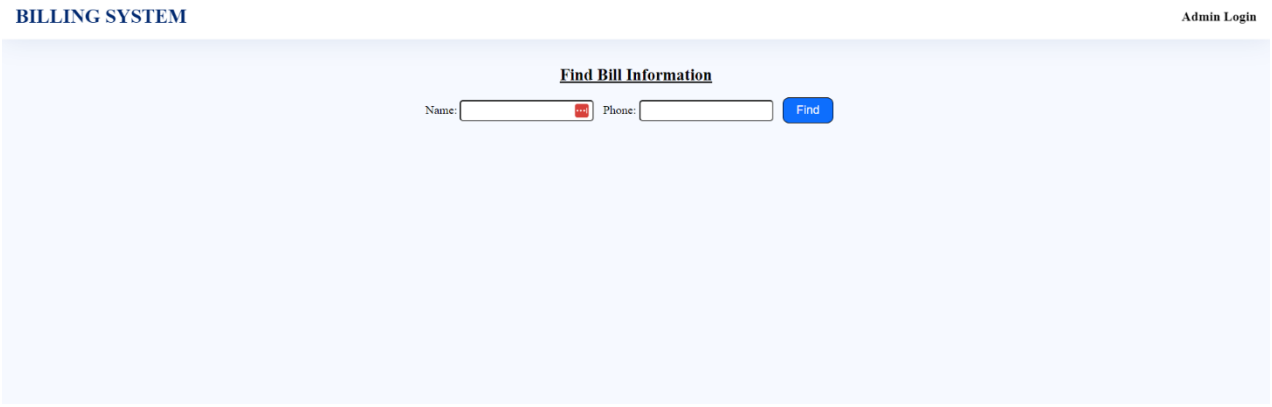

Figure 6.1: Home Page

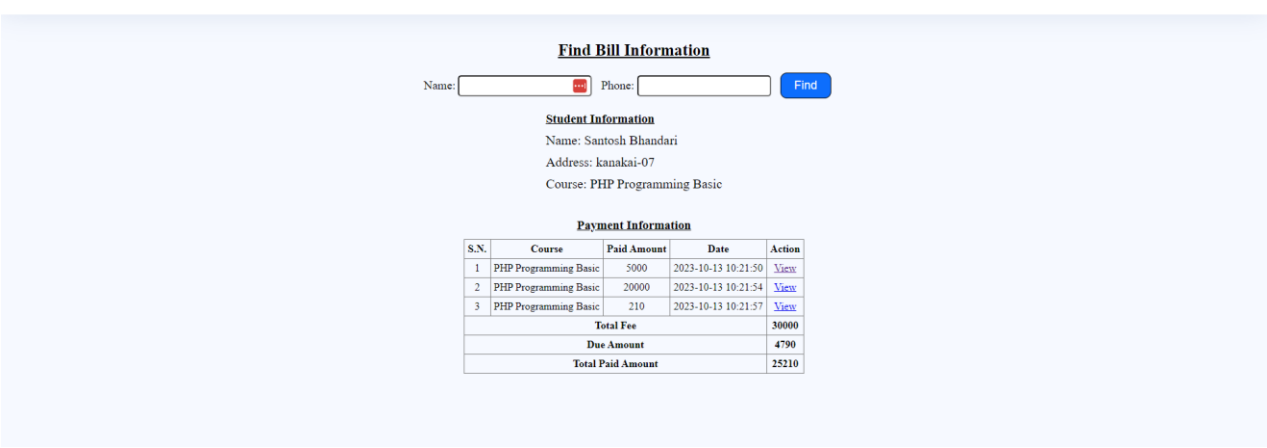

Figure 6.2: Bill Statement

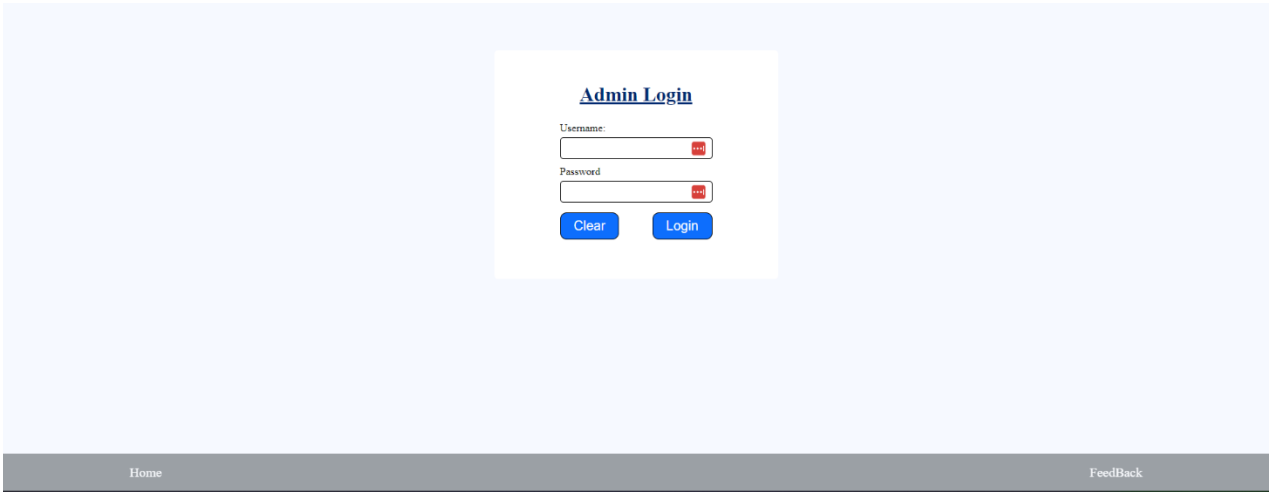

Figure 6.3: Admin Login

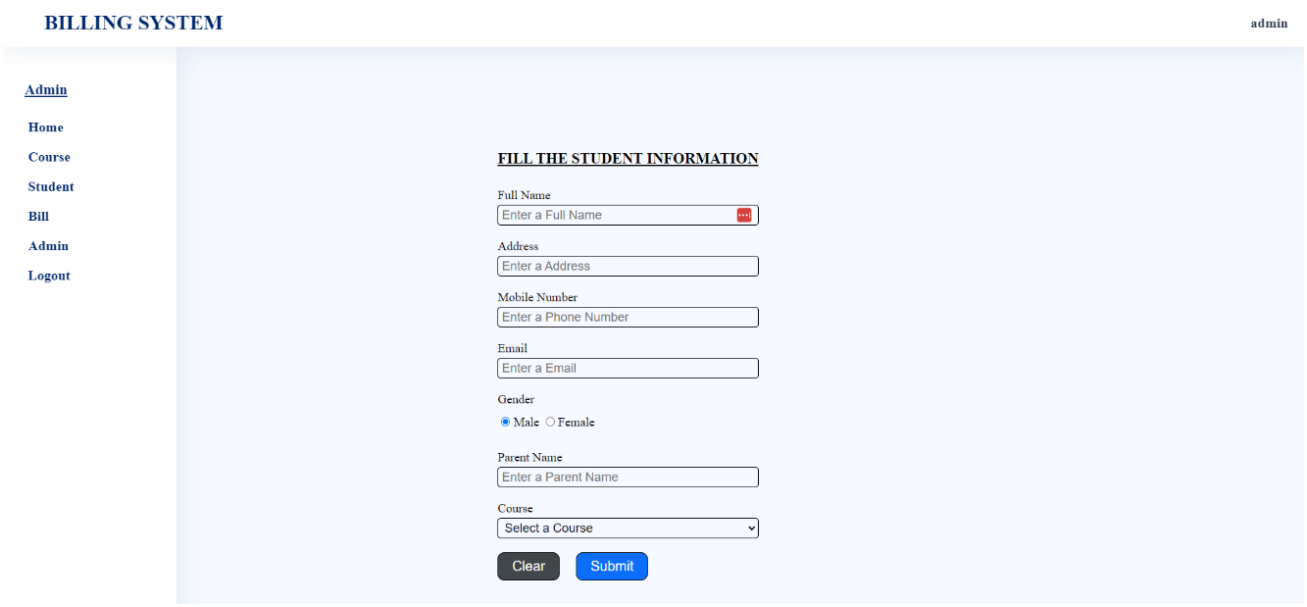

Figure 6.4: Student Information Form

Figure 6.5: Student List

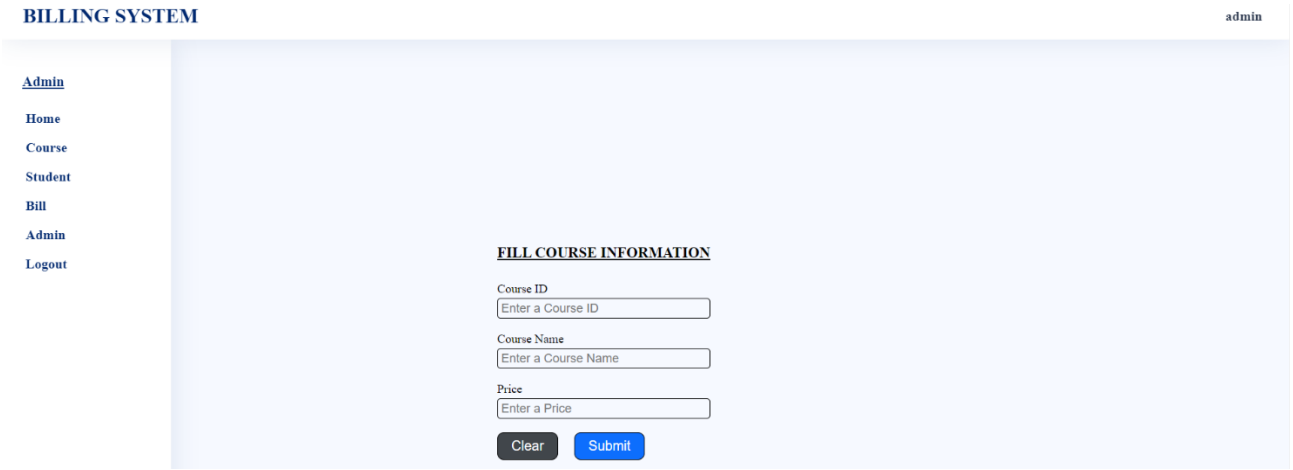

Figure 6.6: Course Information Form

Figure 6.7: Course List

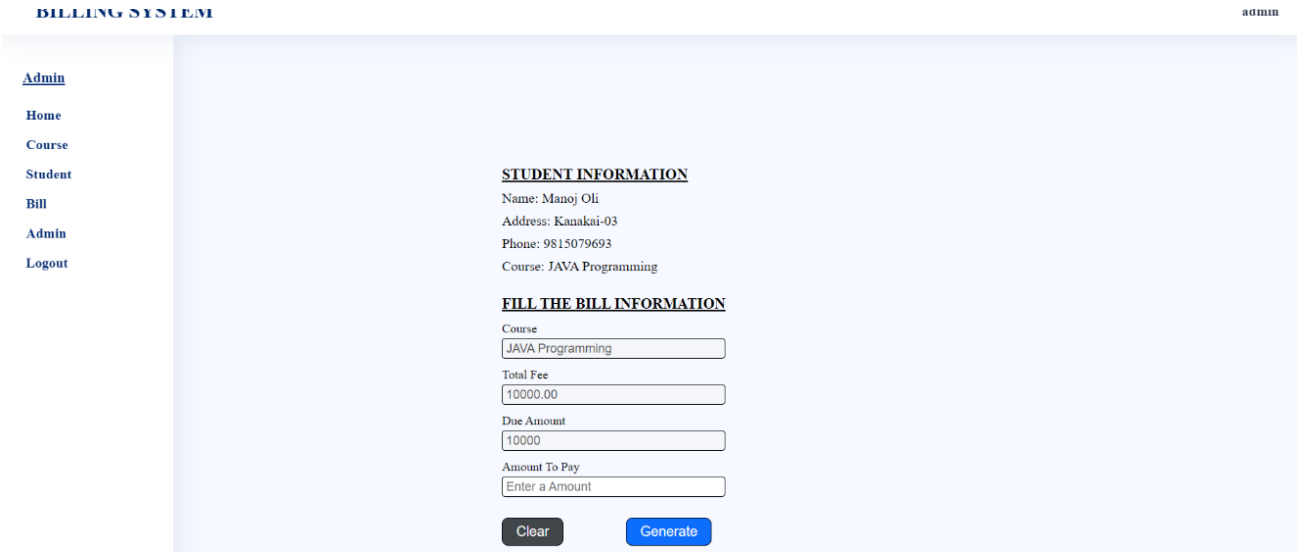

Figure 6.8: Bill Information Form

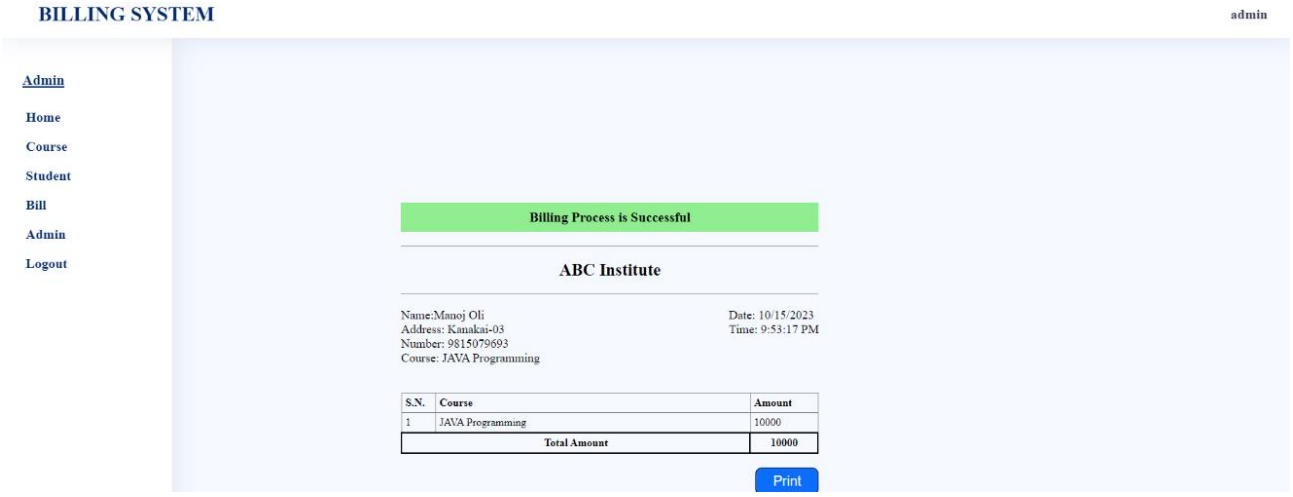

Figure 6.9: Student Bill

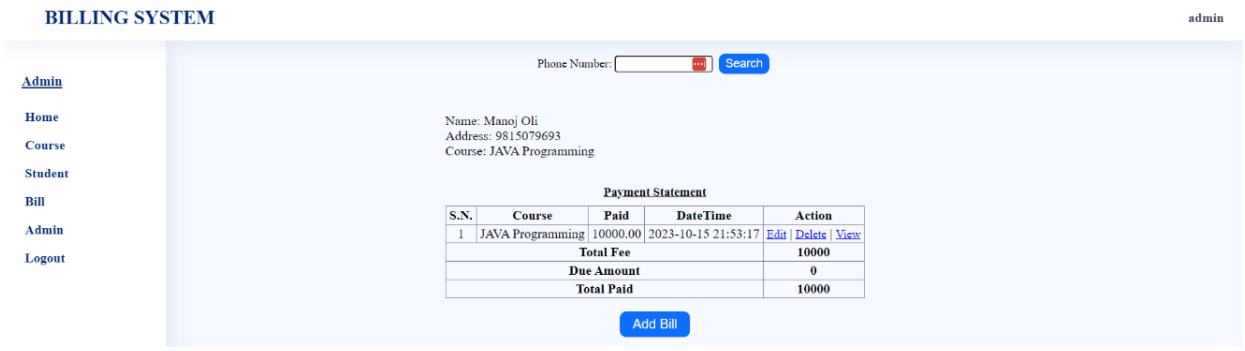

Figure 6.10: Payment Statement

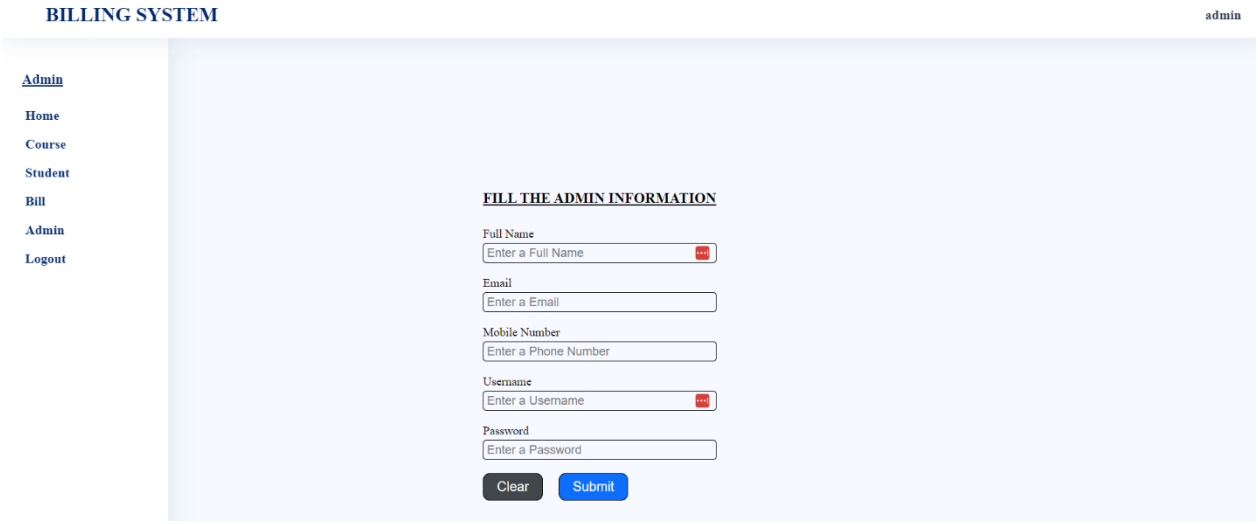

Figure 6: Admin Informaiton Form

Figure 6.12: Admin List

Source Codes

Student Bill Information

```
<?php
    if($_SERVER["REQUEST_METHOD"]=="POST" ||
$_SERVER['REQUEST_METHOD']=='GET'){
        if((isset($_GET['name']) && isset($_GET['phone'])) || (isset($_POST["name"])
&& isset($_POST['phone']))){
            $flag=false;
            if($_SERVER["REQUEST_METHOD"]=="POST"){
                $name=trim($_POST["name"]);
                $phone=trim($_POST["phone"]);
            }
            if($_SERVER['REQUEST_METHOD']=='GET'){
                $name=trim($_GET['name']);
                $phone=trim($_GET['phone']);
            }
            $read="SELECT * FROM StudentInfo,CourseInfo WHERE
StudentInfo.cid=CourseInfo.cid AND name='$name' AND phone='$phone'";
            if($result=$con->query($read)){
                if($result->num_rows>0){
                    $flag=true;
                    while($row=mysqli_fetch_assoc($result)){
                    $name=$row["name"];
                    $address=$row["address"];
                    $course=$row["cname"];
                    $price=$row['price'];
                }
            }
        }
        }
    }
  $read="SELECT * FROM BillInfo WHERE phone='$phone'";
        if($result=$con->query($read)){
            $totalpaidamount=0;
            if($result->num_rows>0){
                $num=0;
                while($row=$result->fetch_assoc()){
                    $num++;
                    $totalpaidamount+=(int)$row['amount'];
                   echo '<tr>
                    <td>'.$num.'</td>
                    <td>'.$course.'</td>
                    <td>'.(int)$row['amount'].'</td>
                    <td>'.$row['tdate'].'</td>
                    <td><a href="view-
bill.php?phone='.$phone.'&date='.$row['tdate'].'">View</a></td>
                    </tr>';
                }
            }
```

```
    }
    echo '
      <tr>
      <td colspan="4"><strong>Total Fee</strong></td>
        <td><strong>'.(int)$price.'</strong></td>
      </tr>
      <tr>
        <td colspan="4"><strong>Due Amount</strong></td>
        <td><strong>'.($price-$totalpaidamount).'</strong></td>
      </tr>
      <tr>
        <td colspan="4"><strong>Total Paid Amount</strong></td>
        <td><strong>'.$totalpaidamount.'</strong></td>
      </tr>
    </tbody>
  </table>';
}
  ?>
```

Admin Login

```
        <?php
            if ($_SERVER['REQUEST_METHOD']=='POST'){
              if(isset($_POST['submit'])){
                $username=trim($_POST['username']);
                $password=md5(trim($_POST['password']));
                if (!$username)
                  echo "<p>* Please enter username</p>";
                 if (!$password)
                  echo "<p>* Please enter password</p>";
                if($username && $password){
                  $read = 'select * from AdminInfo WHERE username
  ='".$username."' and adminpwd='".$password."';';
                  if($result=$con->query($read)){
                    if($result->num_rows>0){
                      $_SESSION["adminname"] = $username;
                      header("Location: admin-dashboard.php");
                    }else{
                      echo "<p>* Invalid username or password.</p>";
                    }
                  }else{
                    echo "error.";
                  }
                }
              }
        }
        ?>
```

Add Course

```
<?php
            if ($_SERVER['REQUEST_METHOD']=='POST'){
              if(isset($_POST['submit'])){
                $cid = trim($_POST['cid']);
                $cname = trim($_POST['cname']);
                $price= trim($_POST['price']);
                $read="SELECT * FROM CourseInfo WHERE cid='$cid';";
                if($result=$con->query($read)){
                  if($result->num_rows>0){
                    echo '<div class="message"><p class="failure">Course
Already Added!!</p></div>';
                  }else{
                    $insert = "INSERT  INTO CourseInfo
values('$cid','$cname',$price);";
                    if($con->query($insert)){
                      echo '<div class="message"><p class="success">Course
Added Successfully!!</p></div>';
                      header("Refresh: 5; URL=course-list.php");
                    }else{
                      echo '<div class="message"><p class="failure">Failed To
Add Course!!</p></div>';
                    }
                  }
                }
              }
            }
        ?>
```

Course List

```
        <?php
                  $read= "SELECT * FROM CourseInfo ORDER BY cid;";
                  if ($result=$con->query($read)) {
                    $num=0;
                    while ($row=$result->fetch_assoc()){
                      $num++;
                      $disp="<tr><td>$num</td>
                      <td>".$row['cid']."</td>
                      <td>".$row['cname']."</td>
                      <td>".$row['price']."</td>
                      <td><a href='edit-course.php?cid=".$row['cid']."'> Edit
</a> | <a href='delete-course.php?cid=".$row['cid']."' onclick='return
Check()'>Delete</a></td>";
                      echo $disp;
                    }
                  }else{
                    echo "Error";
                  }
                ?>
```

Add Student

```
<?php
    if ($_SERVER['REQUEST_METHOD']=='POST'){
      if(isset($_POST['submit'])){
        $name=trim($_POST['name']);
        $address=trim($_POST['address']);
        $phone=trim($_POST['phone']);
        $email=trim($_POST['email']);
        $gender=trim($_POST['gender']);
        $parent=trim($_POST['parent']);
        $cid=trim($_POST['course']);
        if(!$email){
          $email = "NULL";
        }
        $read="SELECT * FROM StudentInfo WHERE phone='$phone'";
        if($result=$con->query($read)){
          if($result->num_rows>0){
            echo '<div class="message"><p class="failure">Phone
Number Already Added!!</p></div>';
          }else{
            $insert = "INSERT  INTO StudentInfo
values('$phone','$name','$address','$email','$gender','$parent','$cid');";
            if($con->query($insert)){
              echo '<div class="message"><p class="success">Student
Record Added Successfully!!</p></div>';
            }else{
              echo '<div class="message"><p class="failure">Failed To
Add Student Record!!</p></div>';
            }
          }
        }
      }
    }
  ?>
```

Student List

```
 <?php
    $read = "SELECT * FROM StudentInfo,CourseInfo WHERE
StudentInfo.cid=CourseInfo.cid ORDER BY StudentInfo.name;";
          if ($result=$con->query($read)) {
            $num=0;
            while ($row=$result->fetch_assoc()){
              $num++;
              $disp="<tr><td>$num</td>
              <td>".$row['name']."</td>
              <td>".$row['address']."</td>
              <td>".$row['gender']."</td>
              <td>".$row['email']."</td>
              <td>".$row['phone']."</td>
              <td>".$row['parentname']."</td>
```

```
                        <td>".$row['cname']."</td>
                        <td><a href='student-
bill.php?phone=".$row['phone']."'>Bill</a> | <a href='edit-
student.php?phone=".$row['phone']."'>Edit</a> | <a href='delete-
student.php?phone=".$row['phone']."' onclick='return Check()'>Delete</a></td>";
                        echo $disp;
                    }
                }else{
                    echo "Error";
                }
            ?>
```

Add Bill

```
<?php
   if($_SERVER['REQUEST_METHOD']=='POST'){
      if(isset($_POST["phone"])){
         $phone=trim($_POST["phone"]);
         $read="SELECT * FROM StudentInfo,CourseInfo WHERE
CourseInfo.cid=StudentInfo.cid AND phone ='$phone';";
         if($result=$con->query($read)){
            if($result->num_rows>0){
               while($row=$result->fetch_assoc()){
                  $name=$row["name"];
                  $address = $row["address"];
                  $course=$row['cname'];
                  $cid = $row['cid'];
               }
            }
         }
         $amount = trim($_POST["amount"]);
         $insert="INSERT INTO BillInfo
VALUES('$phone','$cid','$amount',NOW());";
         $flag=true;
         if($con->query($insert)){
            $flag=true;
         }else{
            $flag=false;
         }
      }}
 if($flag){
echo '<div class="message"><p class="success">Billing Process is
Successful</p></div> ';
}else if($flag==false){
echo '<div class="message"><p class="failure">Billing Process Failed!!!</p>
</div>';
 }else{
echo '<div class="message"><p class="failure">Billing Process Failed!!!</p>
</div>';
  }
?>
```

Add Admin

```
<?php
            if ($_SERVER['REQUEST_METHOD']=='POST'){
              if(isset($_POST['submit'])){
                 $name=trim($_POST['name']);
                 $phone=trim($_POST['phone']);
                 $email=trim($_POST['email']);
                 $username=trim($_POST['username']);
                 $password=md5(trim($_POST['password']));
                 $read="SELECT * FROM AdminInfo WHERE
username='$username';";
                 if($result=$con->query($read)){
                    if($result->num_rows>0){
                       echo '<div class="message"><p class="failure">Selected
Admin Username Already Added !!</p></div>';
                    }else{
                       $insert = "INSERT  INTO AdminInfo
values('$username','$name','$email','$phone','$password');";
                       if($con->query($insert)){
                          echo '<div class="message"><p class="success">Admin
Added Successfully!!</p></div>';
                       }else{
                          echo '<div class="message"><p class="failure">Failed To
Add Admin !!</p></div>';
                       }
                    }
                 }
              }
           }
        ?>
```

Admin List

```
 <?php
        $read= "SELECT * FROM AdminInfo ORDER BY name;";
         if ($result=$con->query($read)) {
                $num=0;
                 while ($row=$result->fetch_assoc()){
                          $num++;
                           $disp="<tr><td>$num</td>
                           <td>".$row['name']."</td>
                           <td>".$row['email']."</td>
                           <td>".$row['phone']."</td>
                           <td>".$row['username']."</td>
                           <td><a href='edit-
admin.php?username=".$row['username']."'>Edit</a> | <a href='delete-
admin.php?username=".$row['username']."'>Delete</a></td>";
                           echo $disp;
                       }
                    }else{
                       echo "Error";
                    }
                 ?>
```

References

[1] A. K. I. K. V. V. Baswaraju Swathi, "Implementation of Improved Billing System," 2020.

[2] "IPS Smart Billing - Billing Software," [Online]. Available: http://www.billingmanagementsoftware.net/about.html.

[3] W. Kernochan, "Tech Target," March 2023. [Online]. Available: https://www.techtarget.com/searchitoperations/definition/MIS-management-information-systems.

[4] "My Bill Book," [Online]. Available: https://mybillbook.in/s/supermarket-billing-software/.

[5] S. S. S. P. Ltd, "LinkedIn," 18 July 2023. [Online]. Available: https://www.linkedin.com/pulse/what-invoice-management-system-definition-importance-types.

[6] J. Bridges, "Feasibility Study," 19 4 2023. [Online]. Available: https://www.projectmanager.com/training/how-to-conduct-a-feasibility-study.

[7] "Chat GPT," Open AI, [Online]. Available: https://chat.openai.com/.

[8] "Stack Overflow," [Online]. Available: https://stackoverflow.com/.

[9] "W3 School," [Online]. Available: https://www.w3schools.com/php/.