Tribhuvan University

Institute of Humanities and Social Science

Project Report On

Budget Tracker

System

In partial fulfillment of the requirement for the

Bachelor Degree in Computer Application

Submitted by:

Manjil Khanal [6-2-469-19-2019]

Manika Raut [6-2-469-18-2019]

Submitted To:

D. A. V. COLLEGE

Department of Computer Application

Under the Supervision of

Er. Suman Gupta

TRIBHUVAN UNIVERSITY

FACULTY OF HUMANITIES AND SOCIAL SCIENCES

D.A.V COLLEGE

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by Manjil Khanal and Manika Raut entitled "Budget Tracker**"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

……..

SIGNATURE

Er. Suman Gupta

Lecturer

Faculty of Humanities and Science

D.A.V College

Tribhuvan University

Faculty of Humanities and Social Sciences

D.A.V COLLEGE

LETTER OF APPROVAL

This is to certify that this project prepared by Manjil Khanal and Manika Raut entitled "**Budget Tracker**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

_____ Signature Er. Suman Gupta Lecturer D.A.V College Dhobighat, Lalitpur	_____ Signature Sashi Bhusan Chaturvedi Co-Ordinator D.A.V College Dhobighat, Lalitpur
_____ Signature Internal Examiner:	_____ Signature External Examiner:

ACKNOWLEDGEMENT

We wish to express our sincere thanks to Sudip Adhikari, HOD, of the BCA Faculty, for providing us with all the necessary facilities for the research.

We are also grateful to Er. Suman Gupta, lecturer, in the Department of Bachelor of Computer Application. We are extremely thankful and indebted to her for sharing expertise, and sincere and valuable guidance and encouragement extended to us.

We take this opportunity to express gratitude to all of the Department faculty members for their help and support. We also thank our parents for the unceasing encouragement, support and attention. We are also grateful to our partner/team who supported us through this venture.

We also place on record, our sense of gratitude to one and all, who directly or indirectly, have their hand in this venture.

With Regards

Manjil Khanal

TU Registration No: 6-2-469-19-2019

Manika Raut

TU Registration No: 6-2-469-18-2019

ABSTRACT

In this 21st Century there are a lot of business organization where there is day to day transaction happening in an organization which becomes very difficult to record the transaction of the business. And to solve this problem there is system developed named Budget Tracker System which is website where users or customers can track their daily business transaction in which they can determine total profit and loss of the business in a day wise, monthly wise and yearly wise basis.

Here, in this project user has facility to enter total amounts of goods purchased and expenses from services in expense section and in income section there is total amounts of goods sold and income from services where user has facility to retrieve data day wise, monthly wise and yearly wise section by selecting date in which total expense and total income is shown along with records in table and from there total profit and loss is determined. This project has been built by using HTML, CSS and JavaScript in frontend and in backend PHP and MySQL is used. This document will discuss each and every steps how the system (Budget Tracker) was built and different testing will be done to finalize the system.

List of **Abbreviations**

CSS	Cascading Style Sheet
DBMS	Database Management System
DFD	Data Flow Diagram
HTML	Hypertext Mark-up Language
SQL	Structured Query Language
PHP	Hypertext Pre-Processor
XAMPP	X Apache HTTP Server MySQL PHP Perl Tomcat

Contents

Table of Figures

CHAPTER 1: INTRODUCTION

1.1 Introduction

Here, system named “Budget Tracker” has been build which is website using HTML, CSS, JavaScript, PHP and MySQL, etc. in which user’s or customers have facility to track daily transaction happening in the business organization. They can record total income and expenses happening in a day and retrieve data and can determine total profit and loss in a day wise, monthly wise and yearly wise. And here, revenue can be generated through the use of Google AdSense by displaying banner ads to the customers. The main idea for project came when business was done using Facebook business page. There was business of us named fashion mate Nepal in which Facebook business page name was Fashion Mate Nepal in which Jordan High Copy shoes were sold by purchasing it from manufacturers and used to sell in retail price, in which the business was in partnership and had difficulty in tracking the budget at the end of month. It was very difficult for us to keep track of every transaction happening in the business. The records which used to be written in paper or copy would get lost and was very difficult for determining accurate transaction happening in the business. Sometimes, it would take us headache of day to determine accurate transaction happening in the business. So, to solve this problem this system was build which could keep track of day-to-day transaction happening in the business and plus in others business too. So, that nobody would have the face problems like us in their business in which every business organization could keep track of day-to-day transaction happening in their business easily without any problem in which there is plan to launch this system in the market where this project is built by addressing the problem happening in the market and solving this problem in which every business organization can track their day-to-day transaction like expense, income, etc. happening in their business easily without any problem at cheaper price and without having any problem in which they can also determine total profit and loss happening in their business easily on day-wise, monthly-wise and yearly-wise basis. They can also edit their records too if they want to edit it.

1.2 Features

Some of the features of this system are as listed below:

- User can create their account, update username password and can change their profile information.
- User has facility to record or insert total amounts of goods sold or bought in a day and can insert record of income from services and expense from services too like expense for ads boosting, etc.
- And another feature of this system is that user can retrieve data that is recorded in a day wise, monthly wise and yearly wise and can view them and delete and update the data recorded.
- And by retrieving data they can view total income and total expenses in monthly wise, day wise and yearly wise.
- And at last user can view total profit and loss in the business.

1.3 Problem statement

The main idea of this project came when there was Digital Marketing business of us. Facebook business page named "Fashion Mate Nepal" was run by us where Jordan High Copy shoes was sold by us through online platform by boosting ads. And this business of was in partnership where there were four members including me. And at the end of month, it was very difficult for us to divide the money that came to us and was difficult too for determining total profit and loss for the business. Sometimes, the record kept paper would get lost too. And the calculations were not 100% accurate so by addressing this problem it was decided to build "**Budget Tracker"** system on web based specially focusing for digital marketing business. Where, the users or customers who use this system would not have to face problems like us. They can easily track their daily transaction of their business. And can easily determine total profit and loss in the business. And they can also track their total investment in the business.

1.4 Aim and Objectives

Objectives of this Project are given below: -

- To track daily transaction happening in their business.
- To determine total income and expenses happening in their business.
- To determine total profit and loss of the business in day wise, monthly wise and yearly wise.
- To generate revenue by showing banner ads to the customers through the use of Google AdSense.

1.5 Project scope and Limitation

The main scope of this project is to target large business organization and small business organization like Facebook Business Page where digital marketing business is done. Here, the people who have business will be able to track their daily transaction happening in their business. So, this project mainly focused for business organization where there is a lot of transaction having in the business organization.

This project cannot be used in those area where there are no internet services available as it requires internet to open the website. For e.g.; A small homestay or business organization in most undeveloped rural areas like mustang cannot use this system as it requires internet for opening the website as the system is based on web.

1.6 Project Report Organization

The Project Report has been organized into various chapters.

1.6.1 Background Study and Literature Review:

Here, the background study of the project is done and the study of existing system is done.

1.6.2 System Analysis and Design:

Here, the requirement analysis of system is done in which functional and non-functional requirement study is done and feasibility study is also done and consists of different

data modelling and process modeling diagram which define the design of the system. It also includes architecture design, interface design and database schema.

1.6.3 Implementation and Testing:

Here, how system is implemented is defined and consists all the modules and function of it too and consists of different tools used for programming too and consists of different test cases for Unit Testing.

1.6.4 Conclusion and Future Work:

This part concludes the project in simple we can say about the summary of project too and it also consists of future work that are left to do.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEWS

2.1 Background Study

In this 21st century there are a lot of small and large business organization where business organization like banks have their own software for keeping records of daily transaction happening in their business but small business organization cannot spend huge amount of money for one software which track records of their daily transaction happening in a day. For e.g.: there was digital marking business named “Fashion Mate Nepal” of us from which Jordan high copy shoes were sold from the Facebook business page by boosting ads. In this business there was a lot of transaction happening where product was bought from wholesale market and sold it in retail and there was a lot of transaction happening in this business. And this business was small and could not offer a lot of money for developing a system which could track the daily transaction happening in the business. So, by addressing this problem a system named “**Budget Tracker”** is built for small business organization to solve the problem that is faced by us in which users can use this system in free without paying and can also generate revenue from it through displaying banner ads to the users from Google AdSense and by using monthly subscription option too at very cheap price which becomes ads free too.

2.2 Literature Review

There is already similar like this system under the name of daily expense tracker which is used for project purpose only but not commercially lunched where the function is similar to us but design is completely different from our system and purpose also.

[1]

Other system like this project are as follows:

Expensify

Available on Android and iOS devices, Expensify is great for making expense reports on the go. A must for frequent business travelers, Expensify allows to take photos of

receipts and manually log expenses. When you take a photo of a receipt in Expensify, the app automatically reads the receipt and translates it into a logged expense. In addition, you can organize expenses by categories, like mileage, travel and food. Best of all, Expensify is free for individuals and just $5 a month for teams.

Pricing: Individual use is free for up to 25 receipt scans per month; team pricing for unlimited scans starts at $5 per user per month (after a six-week free trial).

Wally

Wally is a colorful app with lots of great expense-tracking functionality. You can use Wally on an iOS device, and the basic version of the app is free. Like in other expense-tracking apps, you can take photos of receipts directly in Wally or enter expenses by hand and categorize them accordingly. Wally also tracks your income for you and offers you projected savings for each month. With its focus on personal rather than business finance, Wally is a good fit for people who run microbusinesses or side hustles.

Pricing: The basic version is free; the full premium version (Wally Gold) costs $4.99 per month.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis:

System analysis is the process of studying a procedure or business to identify its goal and purposes and create systems and procedures that will efficiently achieve them. In system analysis there comes requirement analysis, data modeling and process modeling.

3.1.1 Requirement Analysis:

Requirement analysis is a software engineering task that bridges the gap between system level software allocation and software design. It provides the system to specify software function and performance indicate software's interface with the other system element and establish constraints that software must meet.

The basic aim of this stage is to obtain a clear picture of the needs and requirement of the end-user and also the organization. Analysis involves interaction between the clients and the analysis. Usually, analysts have to uncover the real needs of the user even if they don't know it clearly. During analysis it is essential that a complete and consistent set of specifications emerge for the system. Here it is essential to resolve the contradictions that could emerge from information got from various parties. This is essential to ensure that the final specification is consistent.

It may be divided into 5 areas of effort:

- Problem recognition
- Evaluation and synthesis
- Modeling
- Specification
- Review

Each Requirement analysis method has a unique point of view. However, all analysis method is related by a set of operational principle. They are:

- The information domain of the problem must be represented and understood.
- The functions that the software is to perform must be defined.
- The behavior of the software as a consequence of external events must be defined.
- The analysis process must move from essential information to implementation detail.

The requirement can be divided into two types:

a. Functional Requirements

Functional requirements are product features or functions those developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. The functional requirement which are required for our project are:

User requirement

- Registration for new user.
- System login for user with username and password.
- Add Expense for services and goods.
- Manage Expense where user can update and delete records of expense.
- View Expense report day-wise, monthly-wise and yearly-wise.
- Add Income from services and goods.
- Manage Income where user can update and delete records of income.
- View Income report day-wise, monthly-wise and yearly-wise.
- User can update profile Information.
- User can update user password.
- System logout for user.

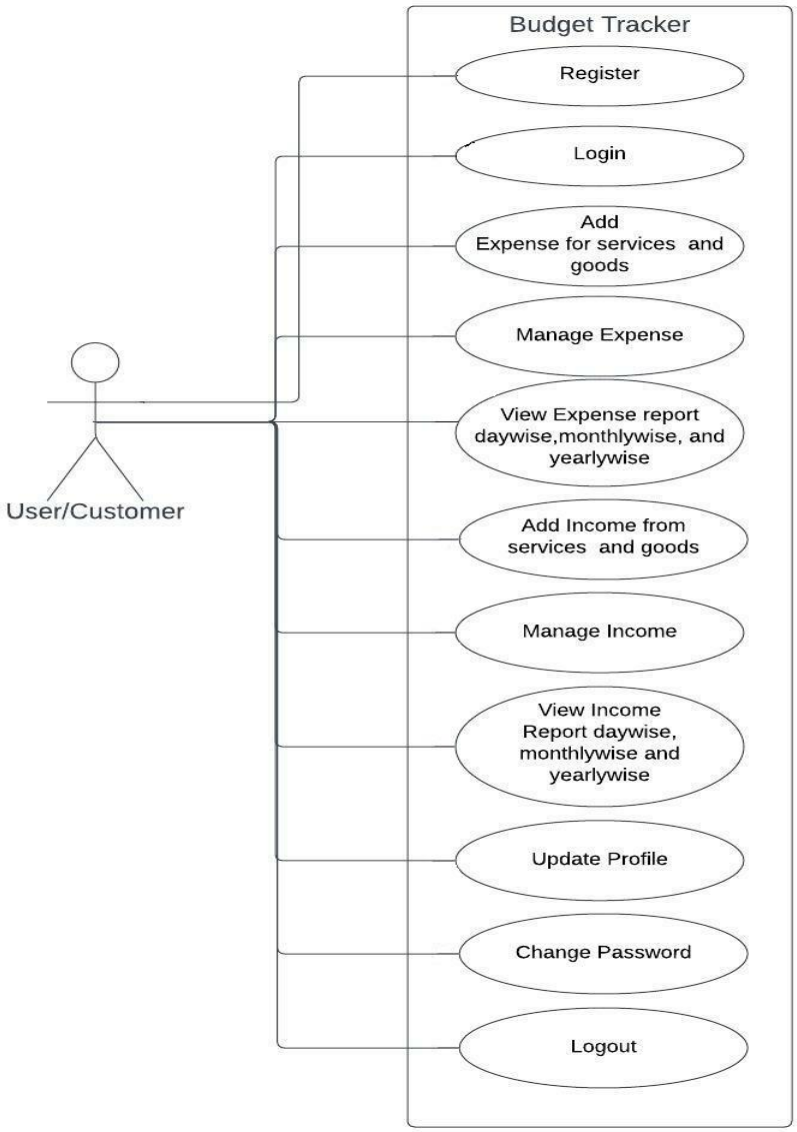

Figure 1: Use Case Diagram

b. Non-Functional requirement

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors. Some non-functional requirements are:

- Usability

Usability defines how difficult it will be for a user to learn and operate the system. Usability can be assessed from different points of view.

- Efficiency of use

It is the average time it takes to accomplish a user's goals, how many tasks a user can complete without any help, the number of transactions completed without errors, etc.

- Security

  Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users' roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information. Security also includes protection against viruses and malware attacks.

- Reliability

  Reliability defines how likely it is for the software to work without failure for a given period of time. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components. To measure software reliability, you can count the percentage of operations that are completed correctly or track the average period of time the system runs before failing.

- Performance

  Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes system safety when it's is overloaded.

- Availability

  Availability is gauged by the period of time that the system's functionality and services are available for use with all operations. So, scheduled maintenance periods directly influence this parameter. And it's important to define how the impact of maintenance can be minimized. When writing the availability requirements, the team has to define the most critical components of the system that must be available at all time. You should also prepare user notifications in case the system or one of its parts becomes unavailable.

- Scalability

  Scalability requirements describe how the system must grow without negative influence on its performance. This means serving more users, processing more

data, and doing more transactions. Scalability has both hardware and software implications. For instance, you can increase scalability by adding memory, servers, or disk space. On the other hand, you can compress data, use optimizing algorithms, etc.

- Final words

  All the software projects include the information boundaries that describe the product and project goals. These boundaries are drawn in the project requirements and specifications. The value of creating a software requirement specification is in the optimization of the development process. Software requirement specifications answer all developer's questions about the product that are required to start the work. The functional specification is approved by the client and ensures that developers are building what the customer wants.

3.1.2 Feasibility Study

Feasibility Analysis is an important phase in the software development process. A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. Feasibility study helps to know the 'strength' and 'weakness 'of the project. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to proceed the project. Here, a lot of study is done for the feasibility of this project. It helped in taking decisions: What kind of project to be made? Which programming language is to be selected? Which database is to be used? What can be the mapping model of objects? Can this project fulfill the requirements of syllabus? If it cannot how it can fulfill? How much time would it take to be completed? What kind of skill do we need? Etc.

Feasibility study should be performed on the basis of various criteria and parameters.

The various feasibility studies are:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Schedule Feasibility

Technical Feasibility:

Technically, this project is a web-based system. All the project was done by us without outsourcing. This project was done by using HTML, CSS, JavaScript, etc. for frontend and backend we used PHP and MySQL, etc. As this project or system is web-based system so this system could be accessed through internet connection by the users. And all, the technology and resources required for developing this system are available in the internet. For this, reason this system or project is technically feasible.

Operational Feasibility:

It refers to the feasibility of the product to be operational. The system will contain simple and easily understandable parts which can and will be targeted for customers. The created project will be GUI based so that everyone can use it easily.

And for using this system it requires internet connection so, the users who have internet connection can use this system. And today internet is almost available in every place except for some rural areas. For, this reason this system or project becomes operationally feasible.

Economic Feasibility:

It refers to the feasibility of the product to be economical. This system can be used free of cost and both revenues also can be generated through the use of banner ads by using Google Ad scene. And for developing this system all work was done by us without outsourcing of work where time of us was only invested but no money was wasted till now. For this reason, this system or project becomes economically feasible.

Schedule Feasibility:

The system has been developed under the particular time period during fourth semester. Fourth semester lasts for 5 to 6 months and the project was completed under the scheduled time. So, the project is also feasible in terms of schedule. Schedule for the development of the project is shown in figure1:

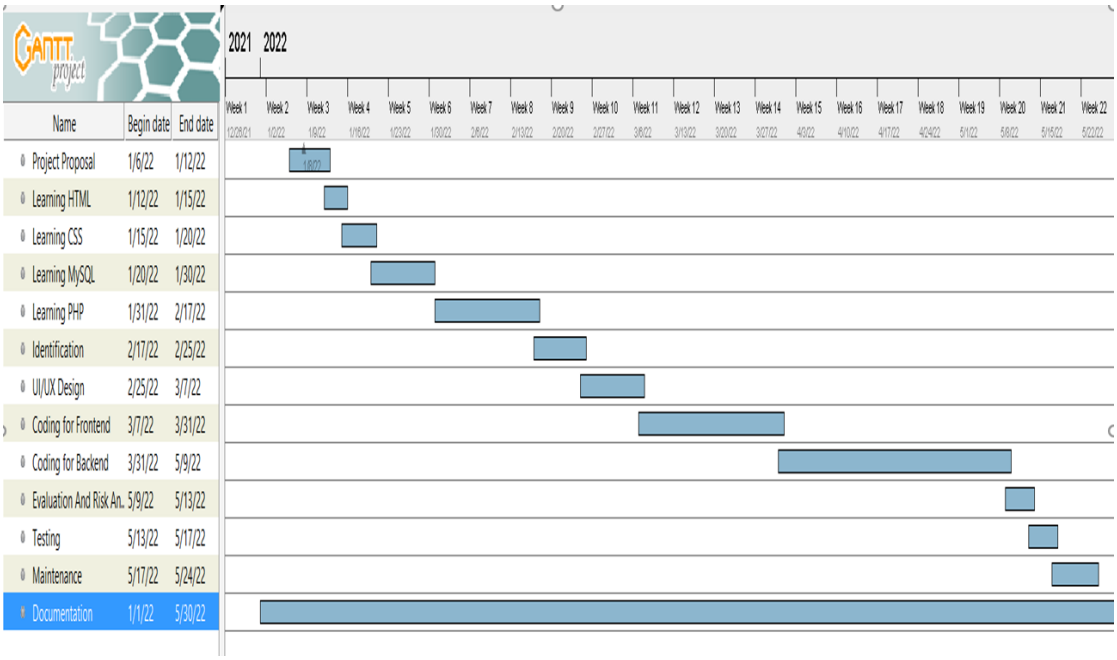

Figure 2: Gantt Chart

3.1.3 Data Modeling

Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system. The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific; hence they were prone to introduce lots of duplication and update anomalies.

Some of the data models for our system are listed below:

ER-Diagram

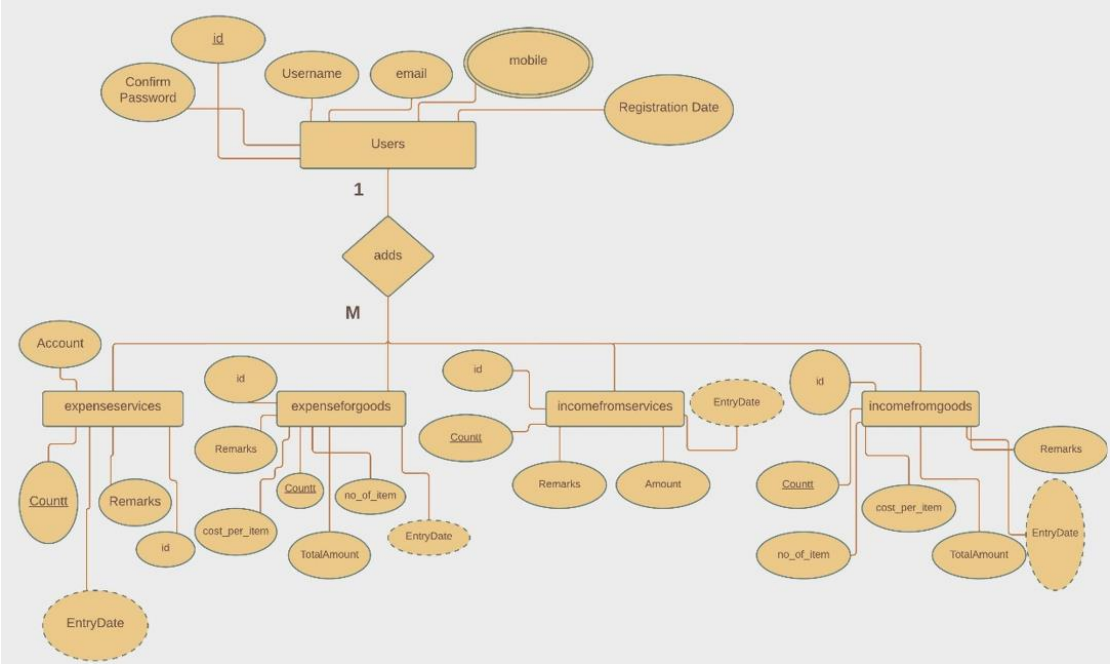

Figure 3: ER-Diagram of Budget Tracker

In ER-diagram first there is user's entity in which users adds expense services, expense for goods, income from services and income from goods, etc. Here, users expense services, expense for goods, income from services and income from goods, etc. becomes entity and adds becomes relationship between them.

Flow chart

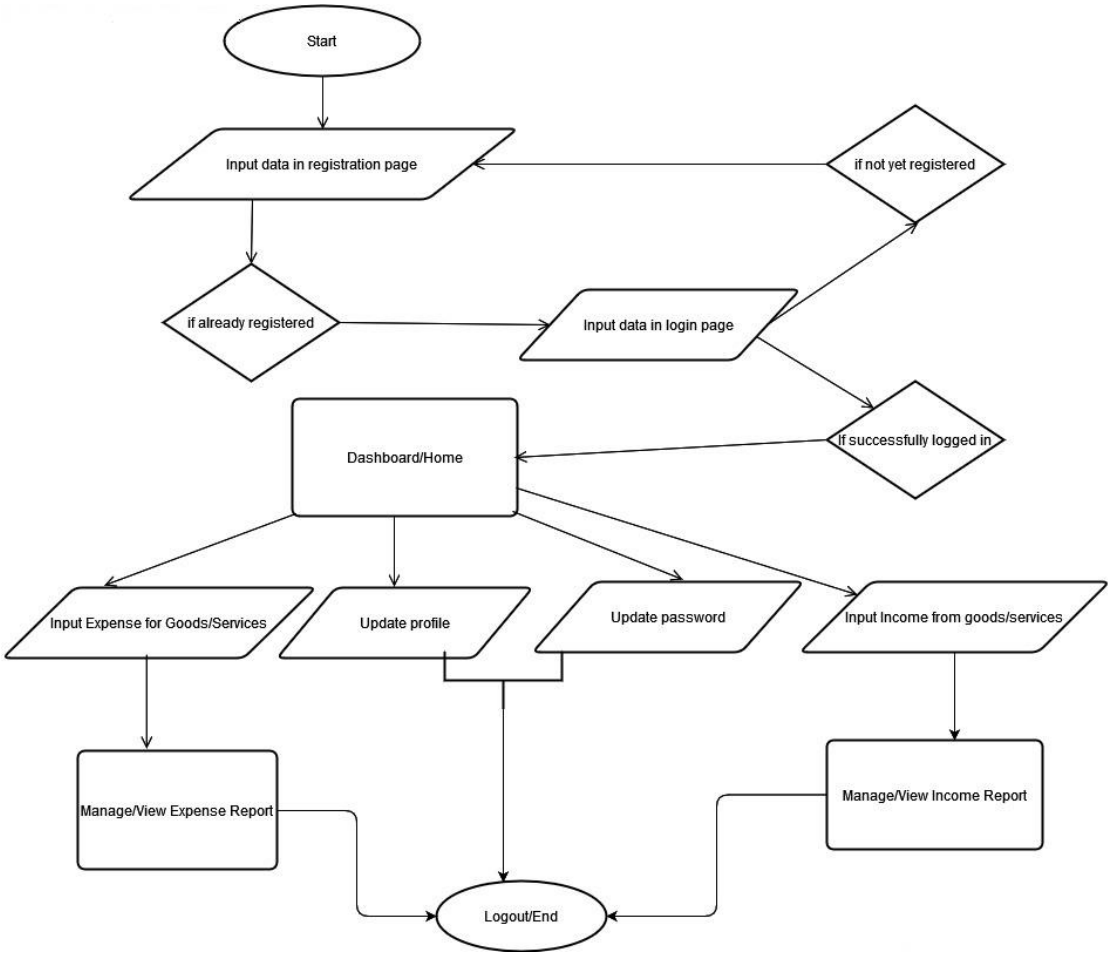

Figure 4: Flowchart

In the above given figure 4, the system first starts and takes input from user for registration and if user is already registered then user is taken to login page otherwise user will be redirected to registration page. After successful login user is taken to dashboard where user can view day-wise, monthly-wise, yearly-wise and till now total income and expenses and from there net profit/loss is determined in which user can see in dashboard. And from dashboard/home page user can input expense for goods/services, input income from goods/services, manage those records, can update profile and can update password and finally can logout and the system ends here.

Context flow diagram

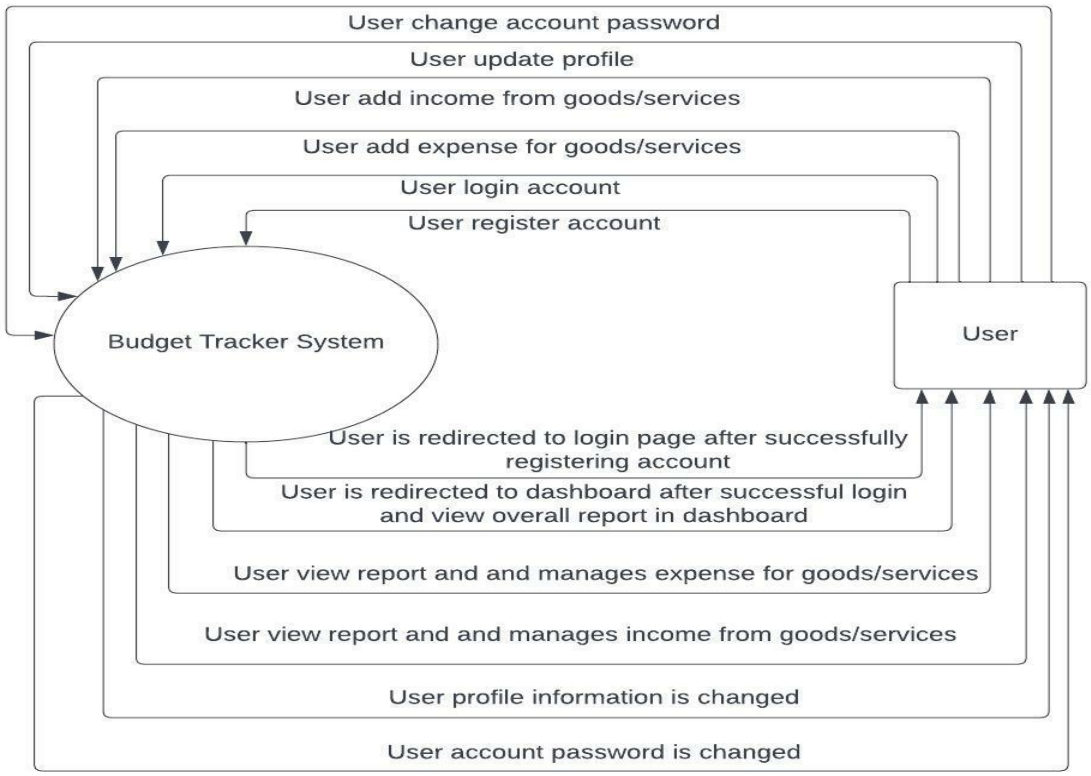

Figure 5: Level 0 CFD (Context Flow Diagram)

In above figure, there is a system named Budget Tracker System and the user uses it by first registering account and after successfully registering user is redirected to login page. User login account and after successful login user is redirected to dashboard where user can view overall report. After that, user add expense for goods/services and user manage and view report of expense for goods/services. And after that, user add income from goods/services and view report of income from goods/services. After that, user can update or updates user profile and user profile information is changed. After that, user changes user account password and user account password are changed. At last, user logout from the system.

Data flow diagram

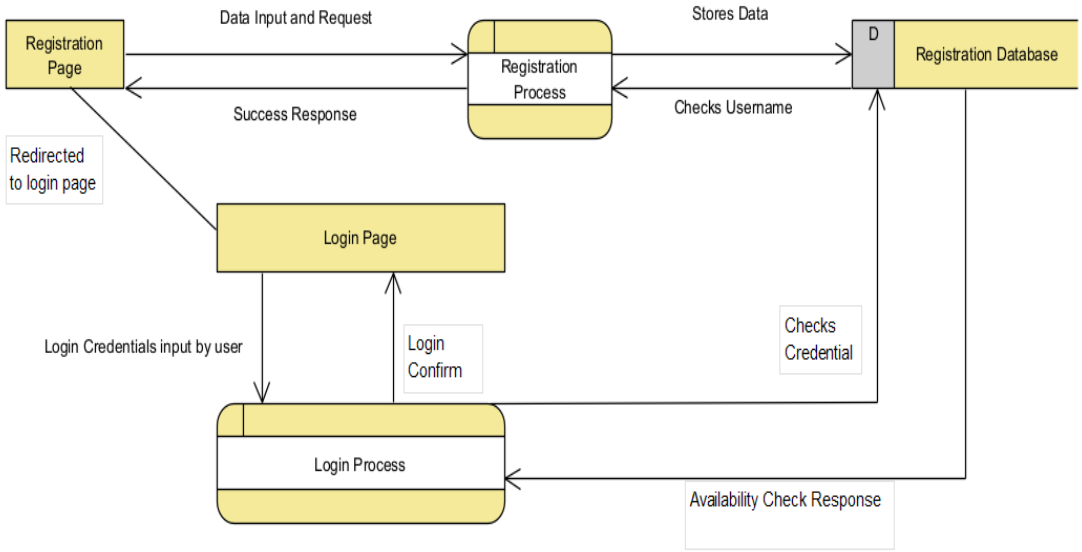

Figure 6: Level 1 DFD for Registration and Login

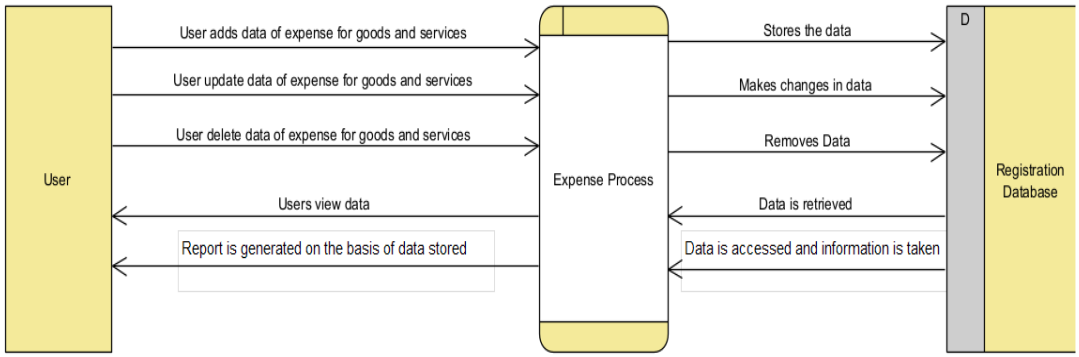

Figure 7: Level 1 DFD for Expense

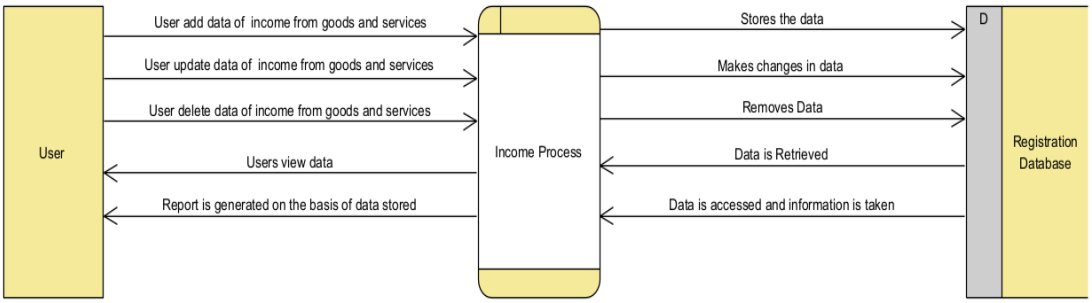

Figure 8: Level 1 DFD for Income

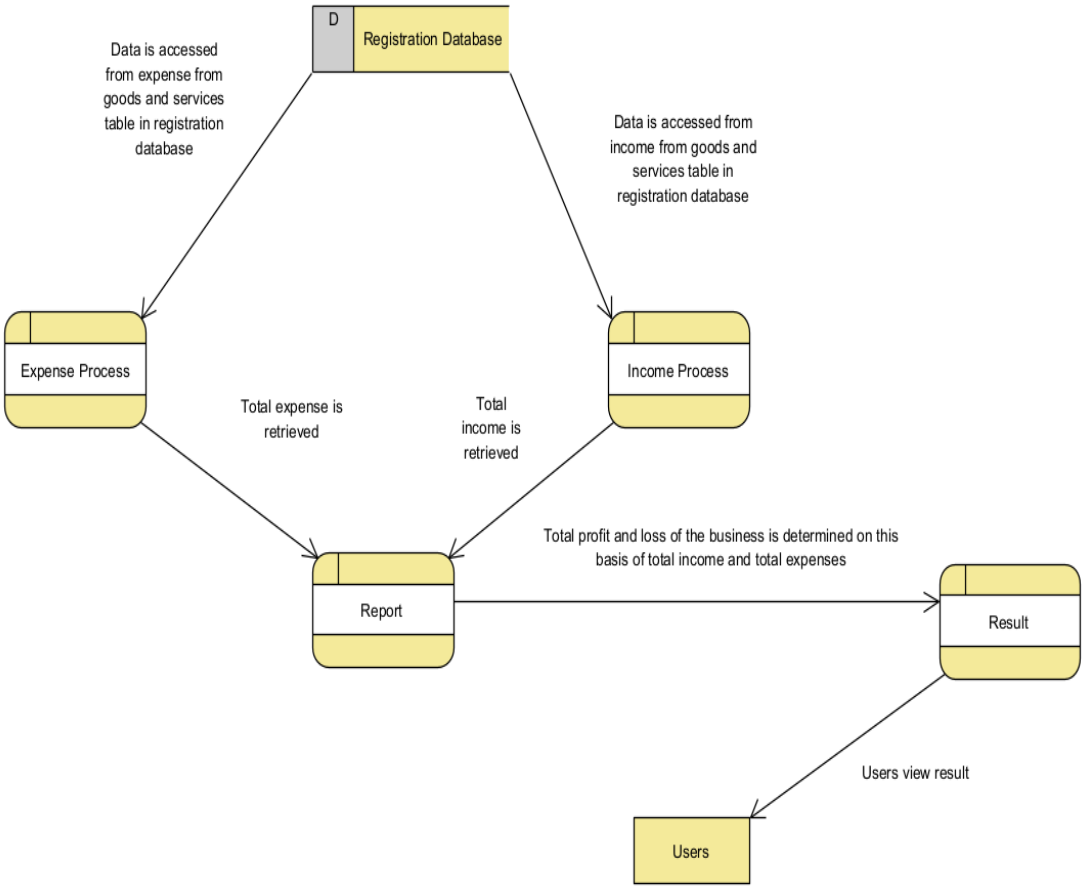

Figure 9: Level 1 DFD Report Viewing

In this Budget Tracker system user will be able to register account and login in the system where user will be redirected to dashboard in which user will be able to view overall report determined from the records. In Expense part, user can add expense for goods and services and after adding user can update and delete the records in manage section and can view the report. In Income part, user can add income from goods and services and after adding user can update and delete the records in manage section and can view the report. At last, total profit and loss in shown in the dashboard which is generated from the total income and expenses table.

3.1.4 Process Modeling

Here, Spiral Model is used for developing the system because there was very less time to complete this project as for Agile Methodology it takes very long time to develop the system. And in this methodology, one of main advantages is that unlike like waterfall model phase can be repeated again and again which is more beneficial for the project.

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the Waterfall model. This spiral model is a combination of iterative development process model and sequential linear development model that is the waterfall model with a very high emphasis on risk analysis. It allows Incremental releases of the product or increment refinement through each iteration around the spiral. The spiral model takes a clue from the iterative model and its repetition. In Spiral model approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. The spiral model passes through four phases over and over in a "spiral" until completely. The four phases are: Identification, Design, Construct or Build and lastly, Evaluation and Risk Analysis.

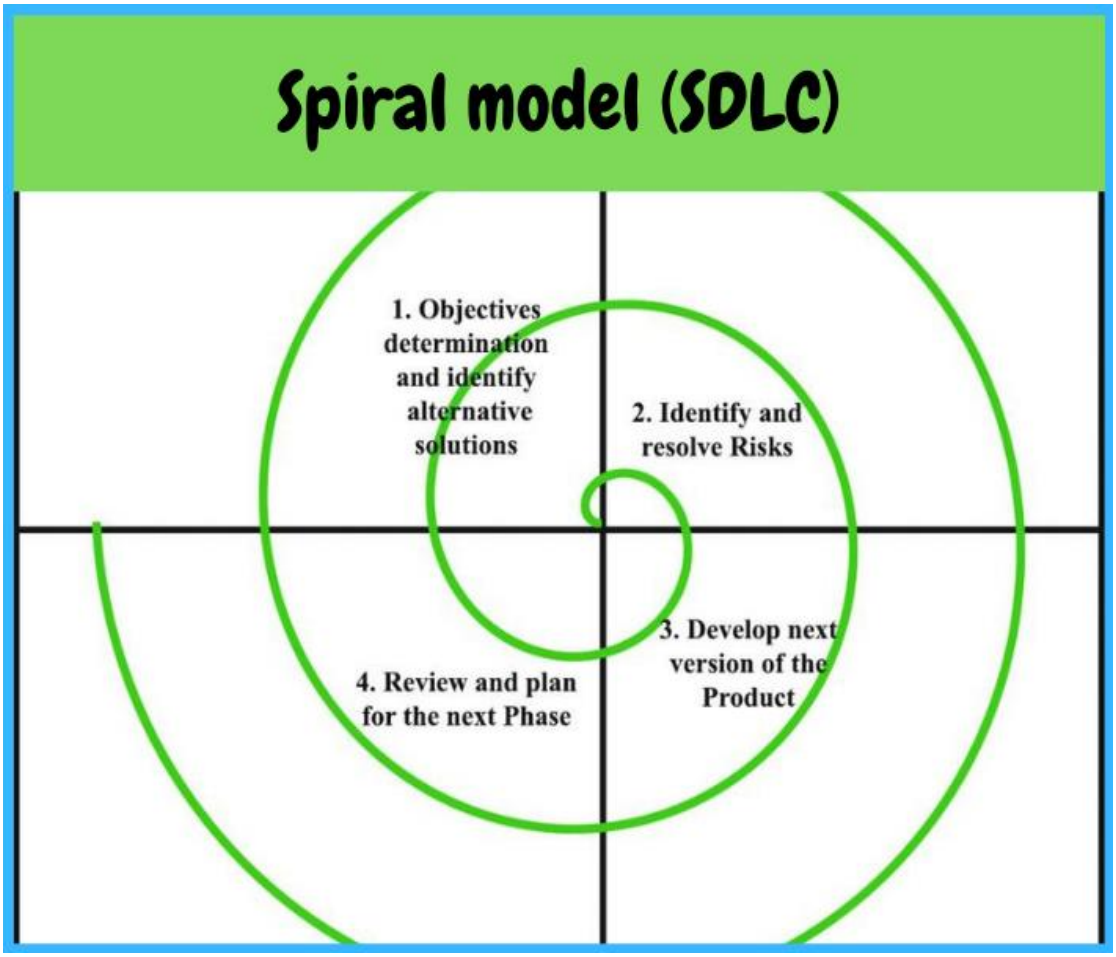

Figure 10: Spiral Model [2]

3.2 System Design

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

3.2.1 Architectural Design

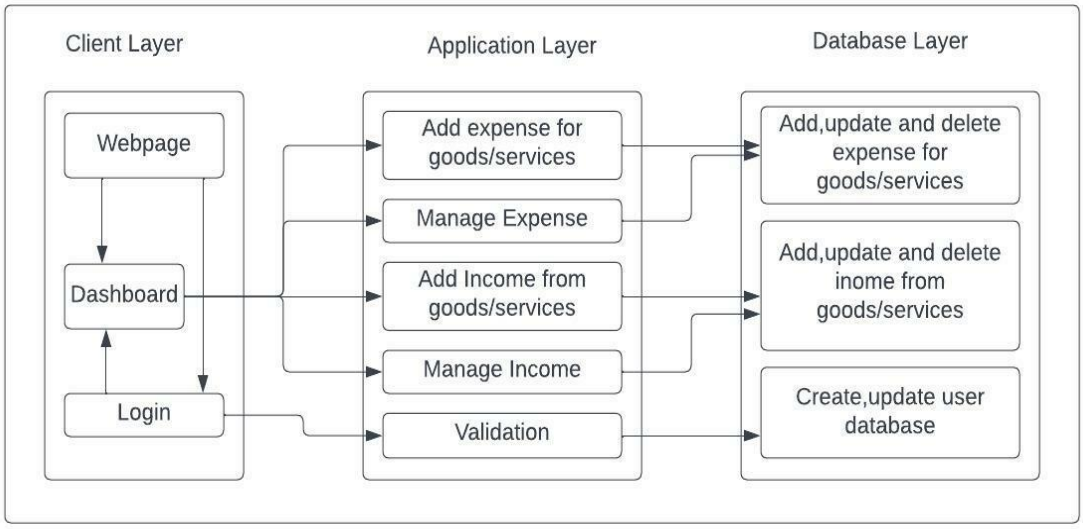

Figure 11: Architectural Design

It is a three based layer model. 3-base architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system.

3.2.2 Database schema

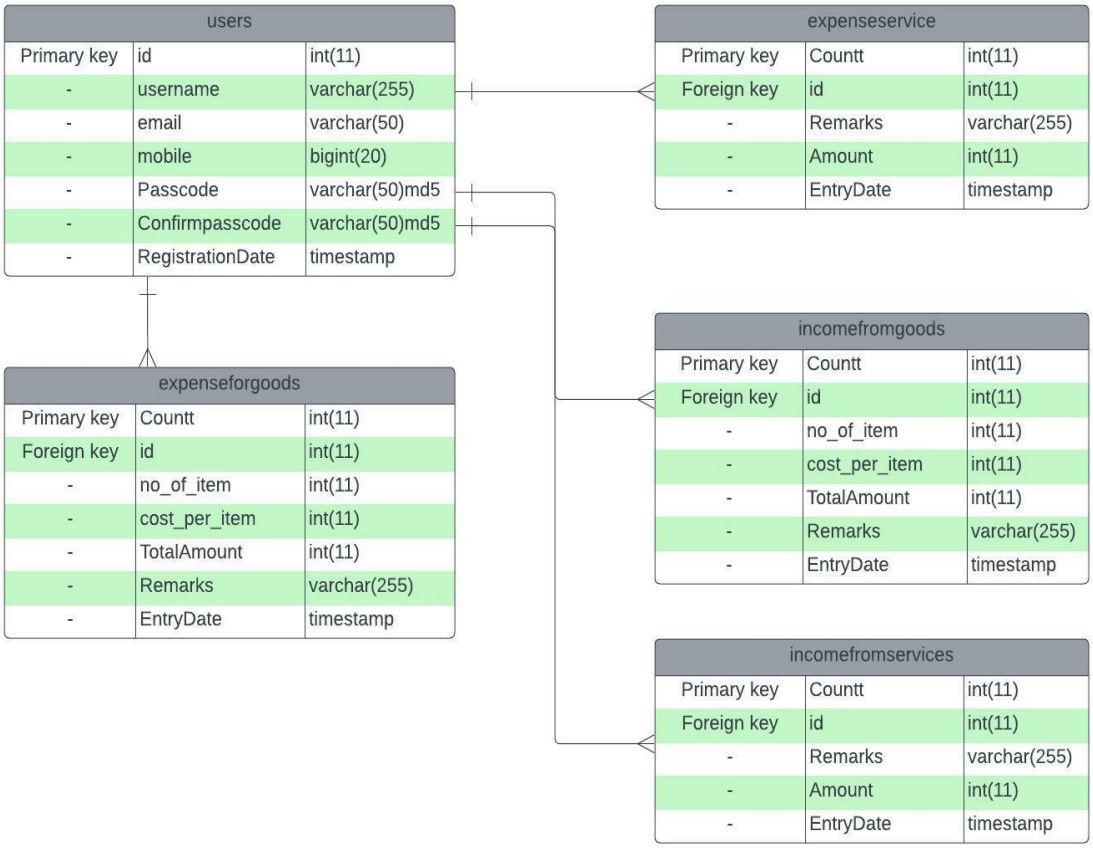

Figure 12: Database Schema

In database schema there are two main word and that is primary key and foreign key. A primary key is used to ensure data in the specific column which is unique and a foreign key is a column or group of columns in a relational database table that provides a link between data in two tables.

In the above database Schema diagram, primary key of each table is connected to another table with the help of foreign key. Each table has their own primary key and the same key is also index in another table so that all the table in database are connected to each other.

3.2.3 Interface design

User interface (UI) design is the process designer use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable. UI design refers to graphical user interfaces and other forms.

Here, Visual studio code is used as an IDE in which HTML, CSS is used in front end and PHP, MySQL and JavaScript in backend.

Registration Page:

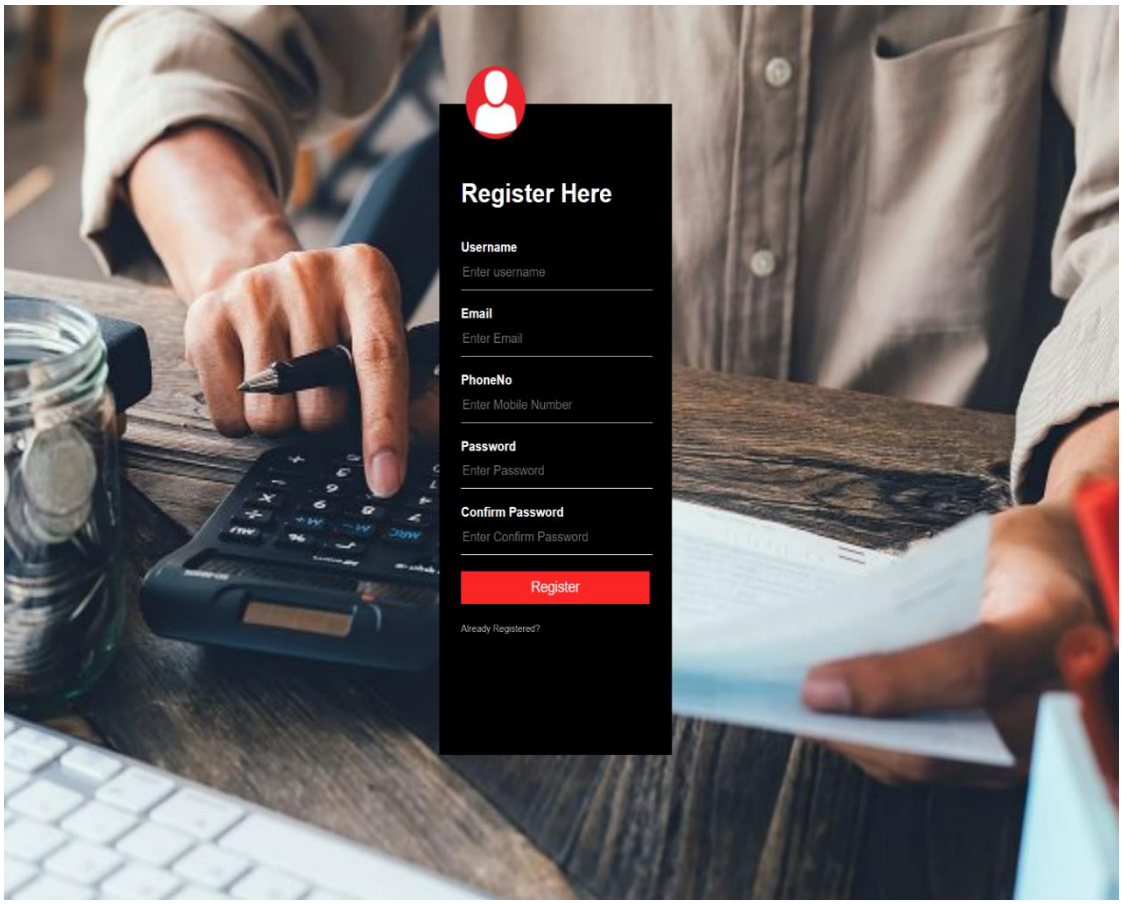

Figure 13: Registration Page

Login Page:

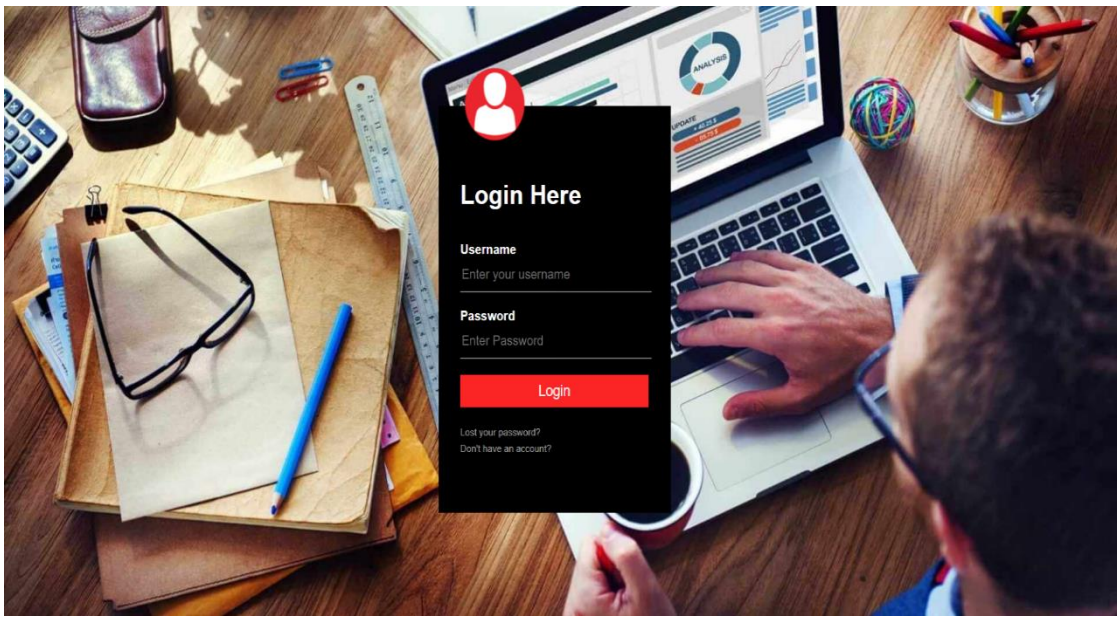

Figure 14: Login Page

Forgot Password Page:

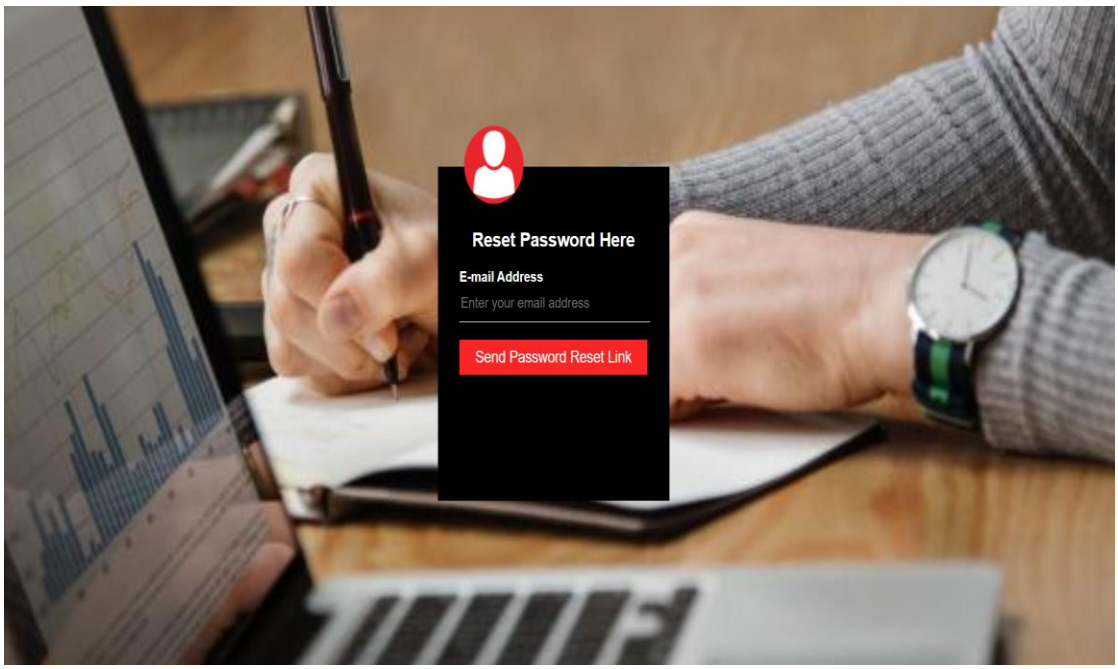

Figure 15: Forget Password Page

Dashboard Page

Figure 16: Dashboard Page

Expense for Services Page

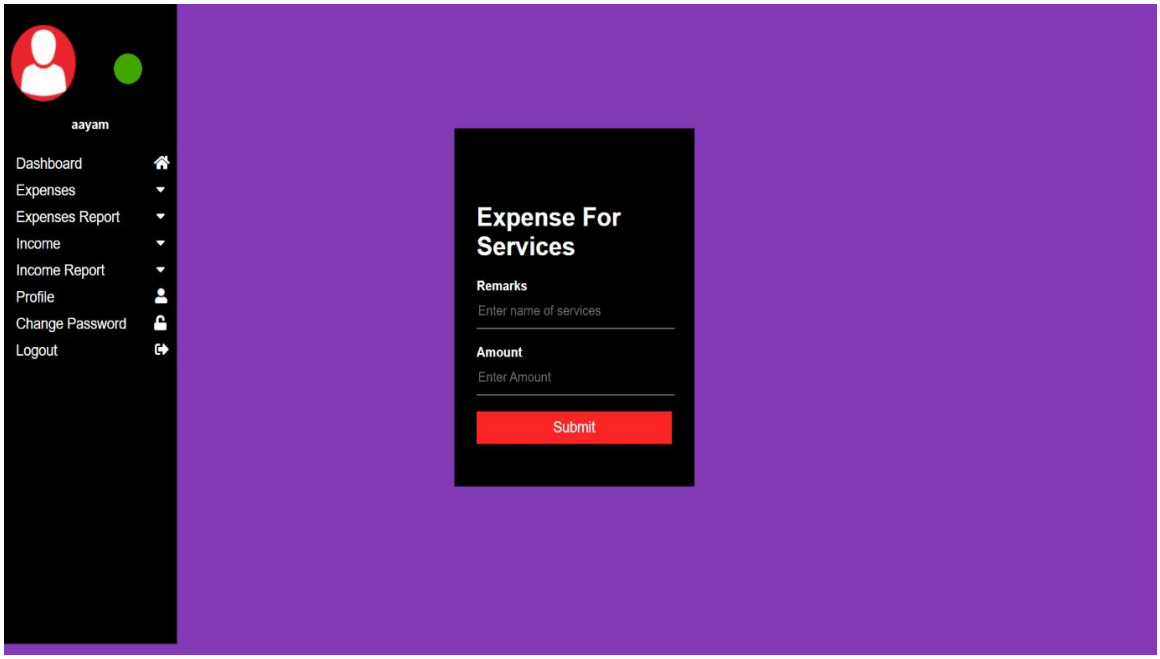

Figure 17: Expense for Services page

Expense for Goods Page

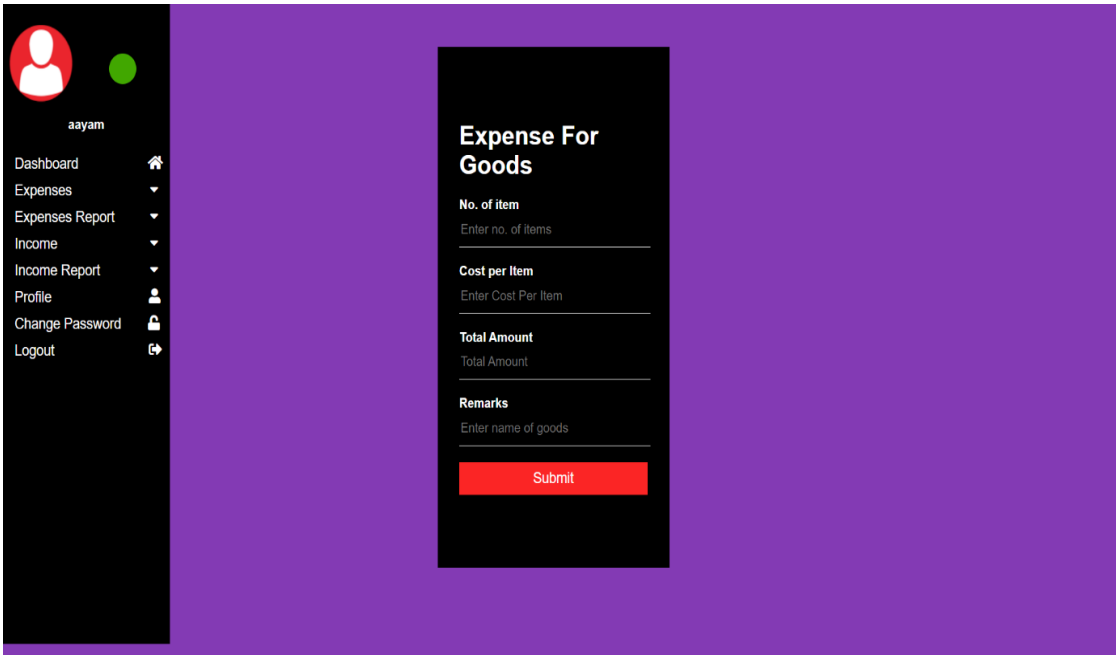

Figure 18: Expense for Goods page

Manage Expense Page

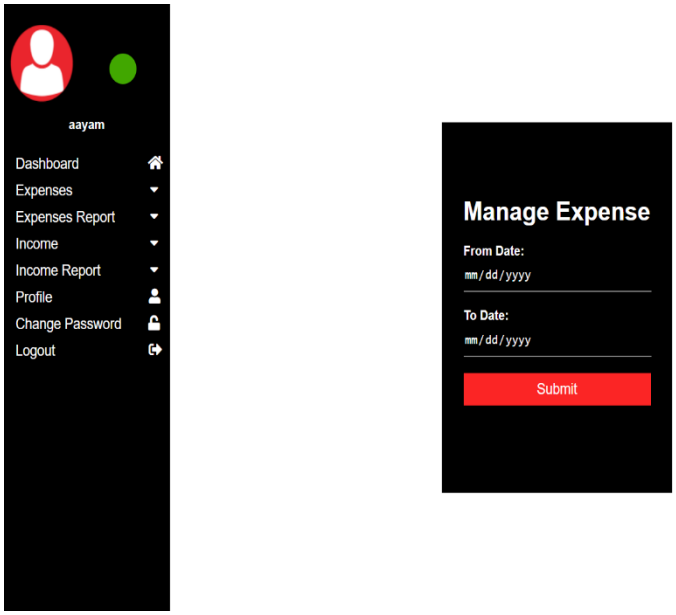

Figure 19: Manage Expense Page

Manage Expense Information Retrieved Page

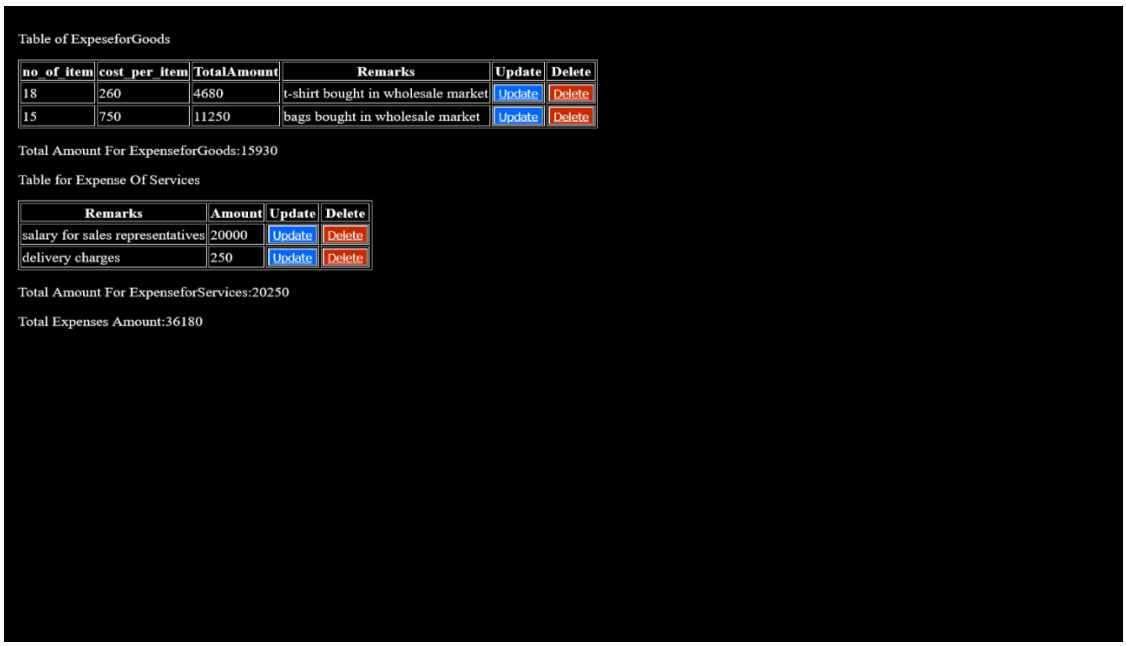

Figure 20: Manage Expense Information Retrieved Page

Day-wise Expense Report Page

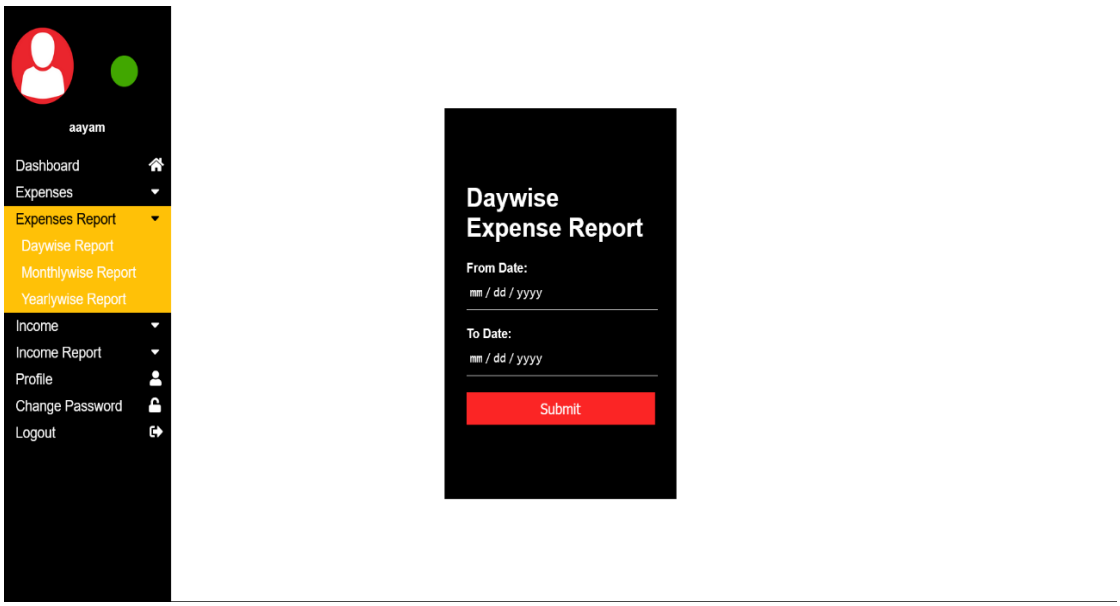

Figure 21: Day-wise Expense Report Page

Monthly-wise Expense Report Page

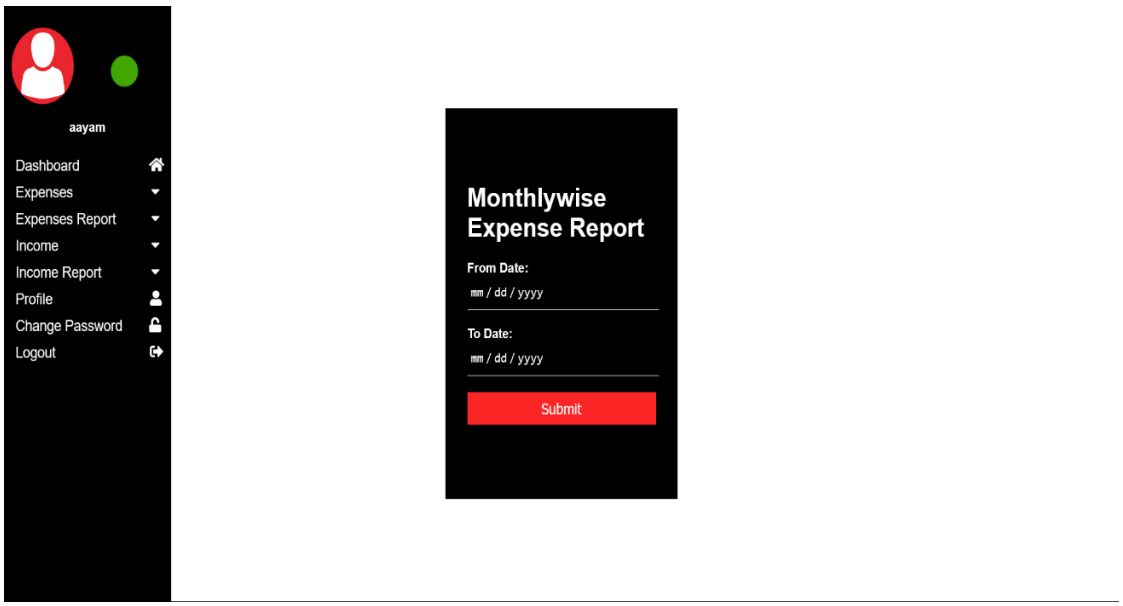

Figure 22: Monthly-wise Expense Report Page

Yearly-wise Expense Report Page

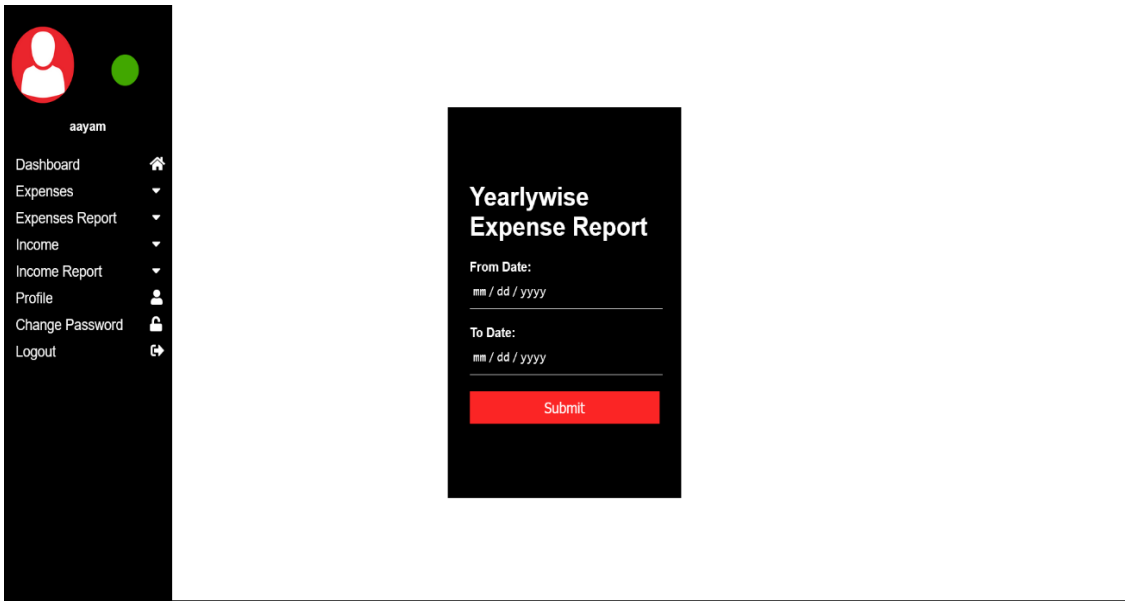

Figure 23: Yearly-wise Expense Report Page

Expense Report Information Retrieved Page:

Figure 24: Expense Report Information Retrieved Page

Income from Services Page

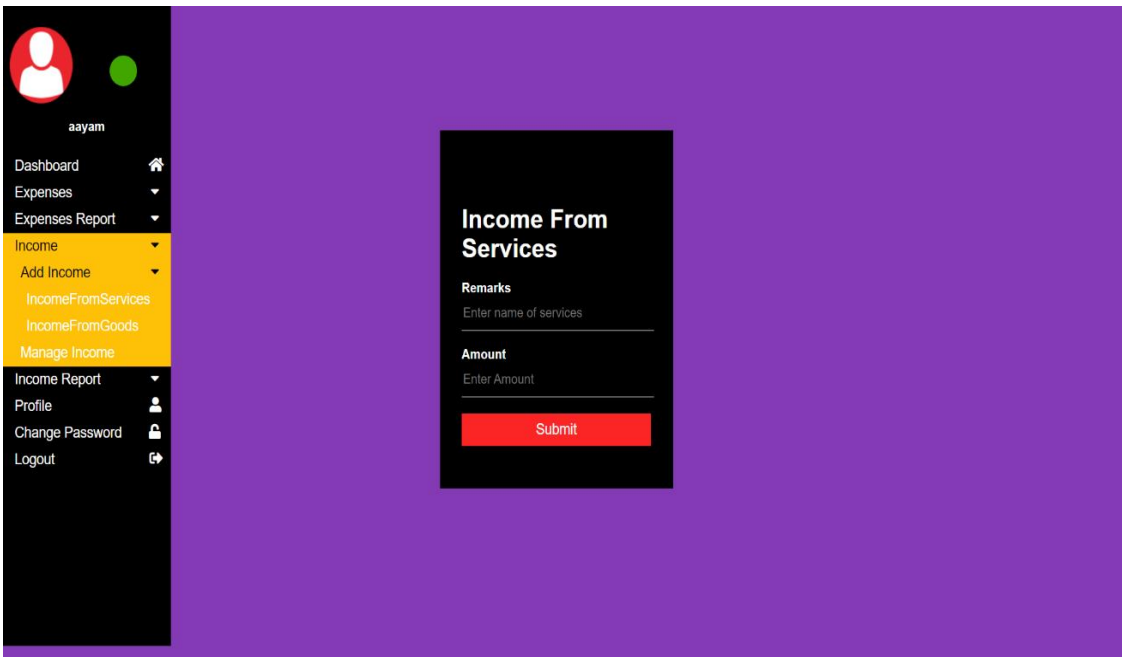

Figure 25: Income from Services Page

Income from Goods Page

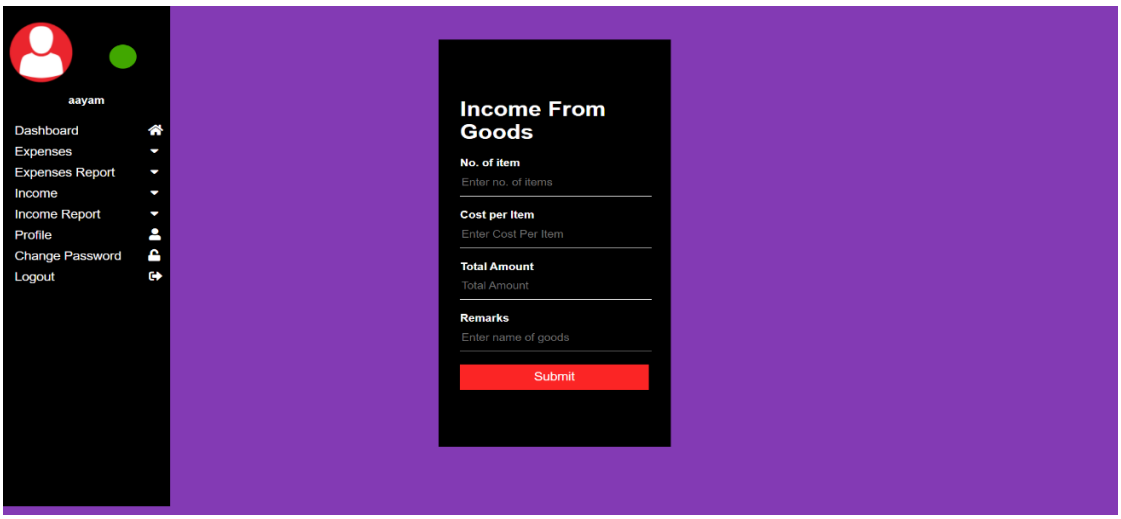

Figure 26: Income from Goods Page

Manage Income Page

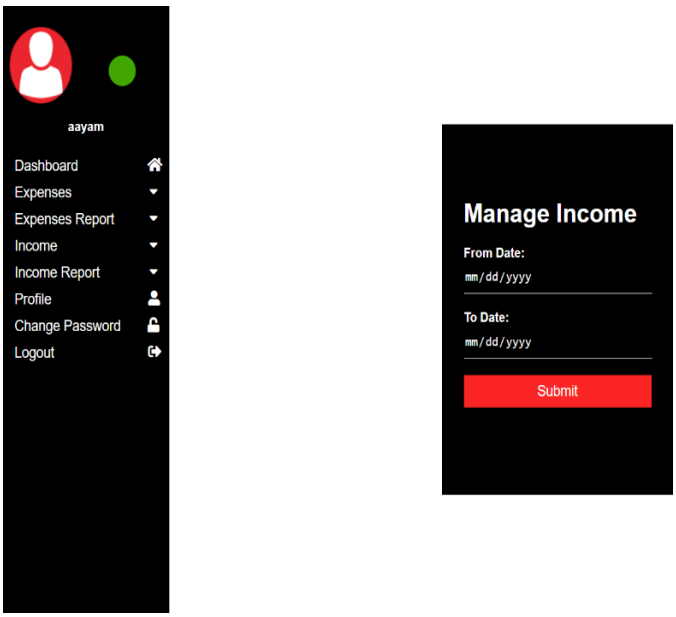

Figure 27: Manage Income Page

Manage Income Information Retrieved Page

Figure 28: Manage Income Information Retrieved Page

Day-wise Income Report

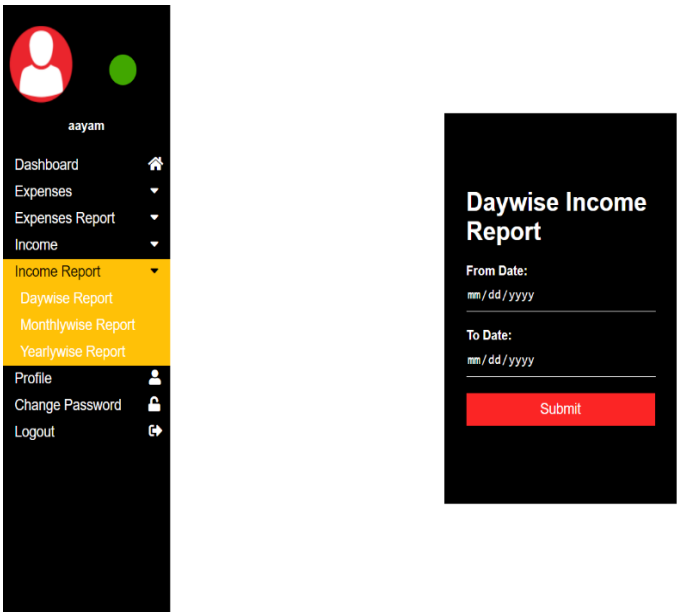

Figure 29: Day-wise Income Report Page

Monthly-wise Income Report Page

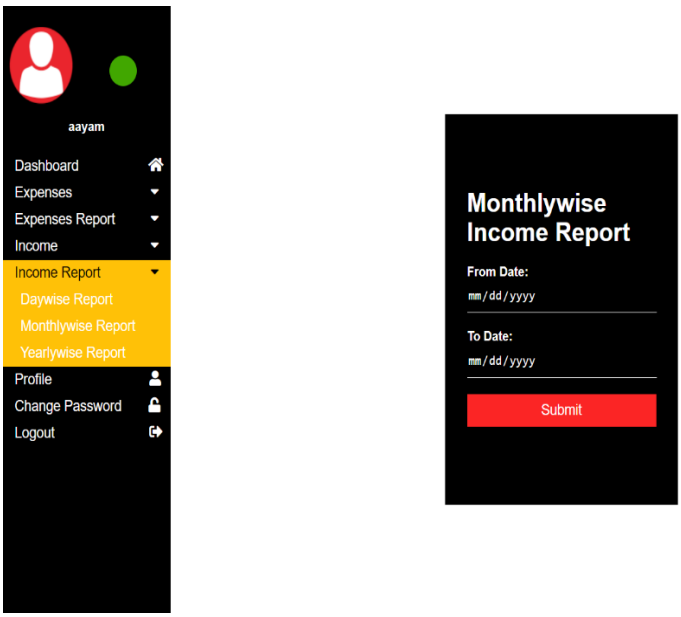

Figure 30: Monthly-wise Income Report Page

Yearly-wise Income Report Page

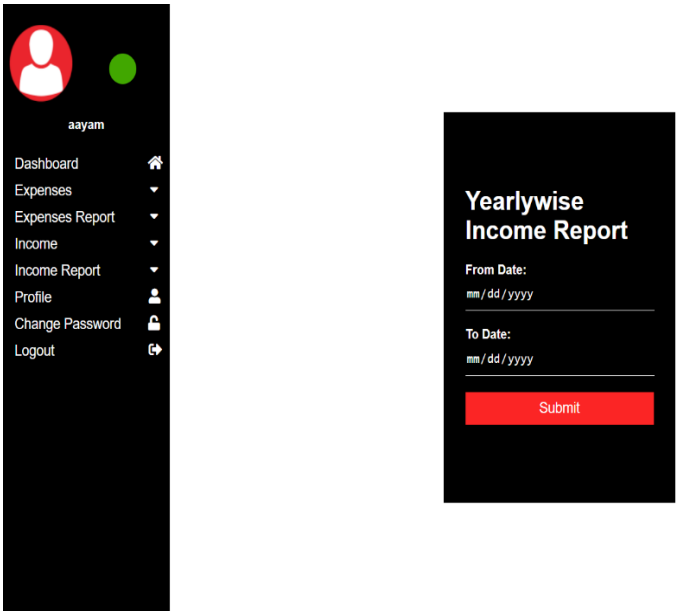

Figure 31: Yearly-wise Income Report Page

Income Report Information Retrieved Page

Figure 32: Income Report Information Retrieved Page

Profile Page

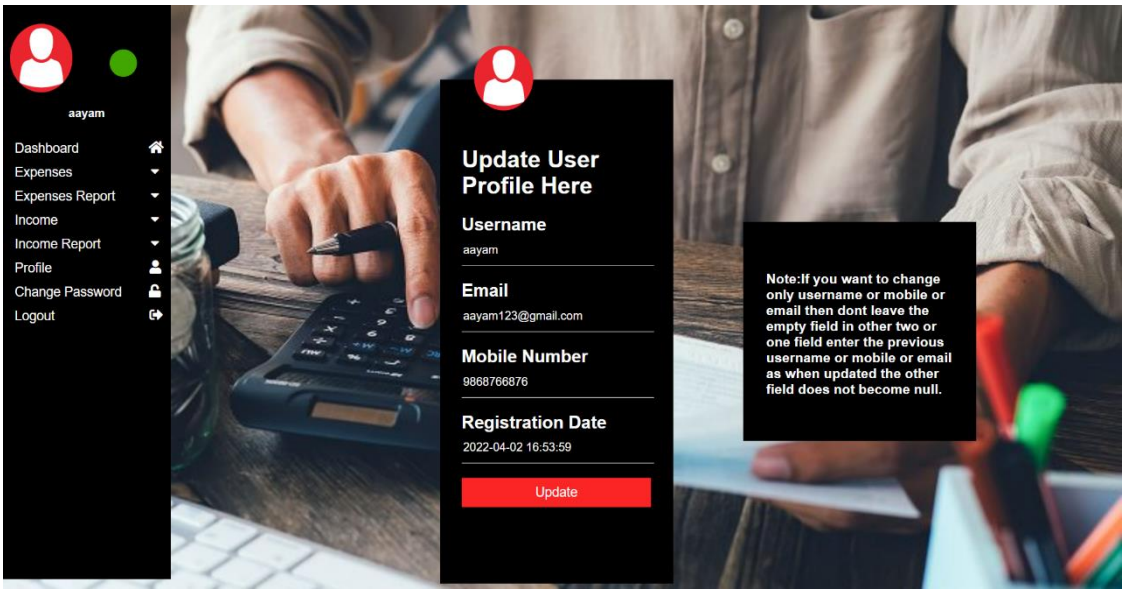

Figure 33: Profile Page

Change Password Page

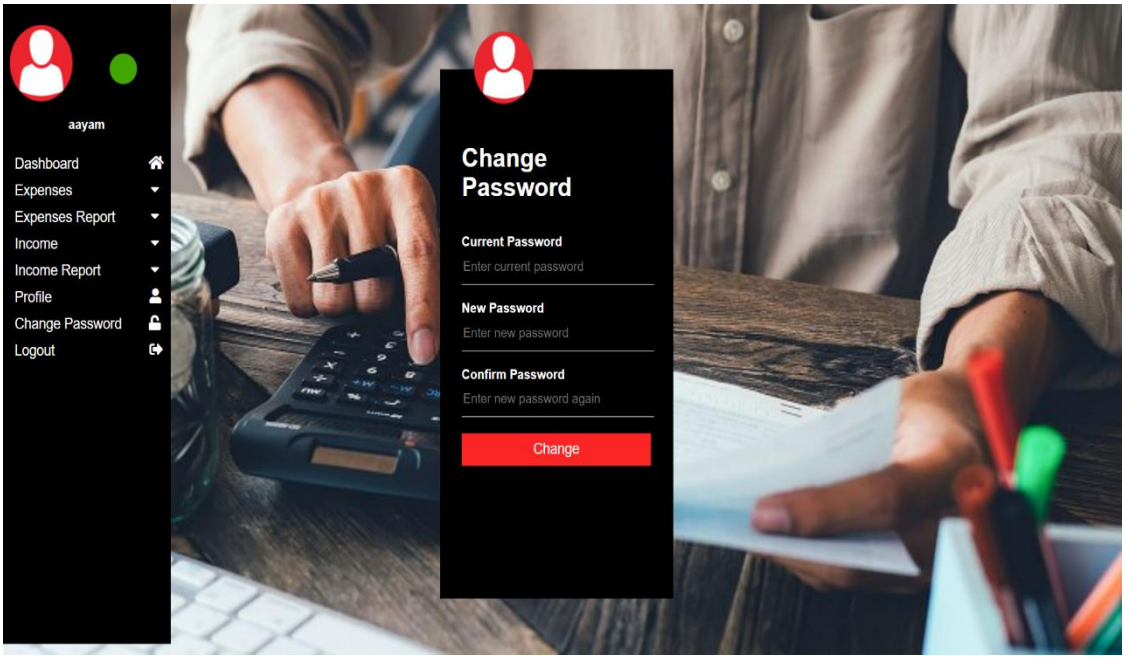

Figure 34: Change Password Page

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Tools Used for Developing the System

Visual Studio Code

Visual Studio Code is used as text editor for coding for developing this system as it easier to do coding in Visual Studio Code as it provides better user-friendly GUI for coding.

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache developed by Apache Friends, consisting mainly of the Apache server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages.
Here, XAMPP is used as local host for storing data of this project and for managing data of this project.

Programming tools

HTML

HTML (Hypertext Markup Language) is used to create document on the World Wide Web. It is simply a collection of certain key words called 'Tags' that are helpful in writing the document to be displayed using a browser on Internet.

Here, in this project HTML is used for defining the structure of this system.

CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

Here, CSS is used in this project for styling the system and for making system more responsive to the users.

JavaScript

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

In this system, JavaScript is used in backend and frontend too. Like in frontend JavaScript is used to show that user is online or offline in the system. And in the backend, JavaScript is used to change report to PDF for downloading it to the user. And another part it is used for determining the total amount in income from goods and income from expenses table.

PHP

PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.

The most part of website in backend is done through PHP like database connectivity, CRUD operation and form validation is done by PHP. It covers maximum backend part of system. So, in this project PHP is used more.

4.2 Implementation detail of Modules

This system is built to make the system which can track the daily income and expenses of business organization happening in a day from which net profit/loss can be determined. Here, the system is based on user/customer it has no admin page in which customer simply create their account through registration page and login to the system through login page and they are redirected to dashboard where user can user can view day-wise, monthly-wise, yearly-wise and till now total income, expense and net profit/loss and from the help side navigation bar they are go or navigate to other page in which after dashboard there comes page for expense section in which user can add expense from goods/services and manage those records by updating and deleting the data of expense which are recorded by user. After that there is expense report button in side navigation bar in which user can retrieve day-wise, monthly-wise and yearly-wise records of expense. After the expense part, there comes button for income in which user can add income from goods/services and can manage those records by updating and

deleting the records of income added by user. After that there comes button below in navigation bar income report through which user can retrieve records of income in a day-wise, monthly-wise and yearly-wise basis according to their need. And after that below there is profile button where by clicking it user is redirected to profile page in which user can update their personal information like mobile no, username, email, etc. After changing their personal information in profile section then below this button in side nav bar there is change password button in which user can change the password of their account. After that, there comes logout button below at last in which user can logout from the system where the session is destroyed.

Some of the main modules of our system are as follows:

- Login registration system.
- Dashboard
- Add and manage expense in the system. (In manage expense part user can update and delete the records of expense)
- Retrieving records of expense in day-wise, monthly-wise, and yearly-wise basis.
- Add and manage income in the system. (In manage income part user can update and delete the records of income)
- Retrieving records of income in day-wise, monthly-wise and yearly-wise basis.
- Updating user profile
- Updating user account password.
- Logout

Login Registration module: This module helps the user to register their account and to login in the system.

Dashboard module: This module helps the user to view overall total income, total expense and total net profit/loss in day-wise, monthly-wise, yearly-wise and till now basis.

Expense Report module: It generates report or data is retrieved in a day-wise, monthly-wise and yearly-wise basis.

Manage Income: It allows user to update and delete the records of income from goods and services.

Income Report module: It generates report or data is retrieved in a day-wise, monthly-wise and yearly-wise basis.

Updating user profile module: It is used to update user profile information like username, mobile no., email, etc.

Updating user account password module: It is used to update user account password.

Logout module: It is used logging out user from the system and is used to destroy the session of user.

4.3 Testing

Software testing is an activity which aims at evaluating the quality of a software product and also to improve it by identifying defects. Software testing strives to achieve its objectives but has certain limitations. However, adherence to the established objectives ensures effective testing.

4.3.1 Purpose of testing

Testing is about verifying that what was specified is what was delivered: it verifies that the product (system) meets the functional, performance, design, and implementation requirements identified in the procurement specifications.

The purpose of testing is to access or evaluate the capabilities or attributes of a software program's ability to adequately meet the applicable standards and application need. Testing can be verification and validation or reliability estimation. The primary objective of testing includes:

- To identify defects in our project "Budget Tracker".
- To check the proper working of the code for adding, managing income and expenses of the goods and services and to check if the session of user is working properly or not and CRUD operation is performed well or not.

4.3.2Test case for unit test

Unit Testing is a software testing technique by means of which individual units of software i.e., group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module is tested to determine if there are any

issue by the developer himself. It is correlated with functional correctness of the independent modules

The objective of Unit Testing are:

- To isolate a section of code.
- To verify the correctness of code.
- To test every function and procedure.
- To fix bug early in development cycle and to save costs.
- To help the developers to understand the code base and enable them to make changes quickly.
- To help for code reuse.

Test Case for Unit Testing:

Test Case 1: User

User Page Registration.

Objective	To register new user account
Action	Fill the information required in the form and click register
Expected Results	Redirected directly to login page after successfully registering the account by clicking on register button
Actual Results	Redirected to login page after successfully registering the account by clicking on register button
Conclusion	Test Successful.

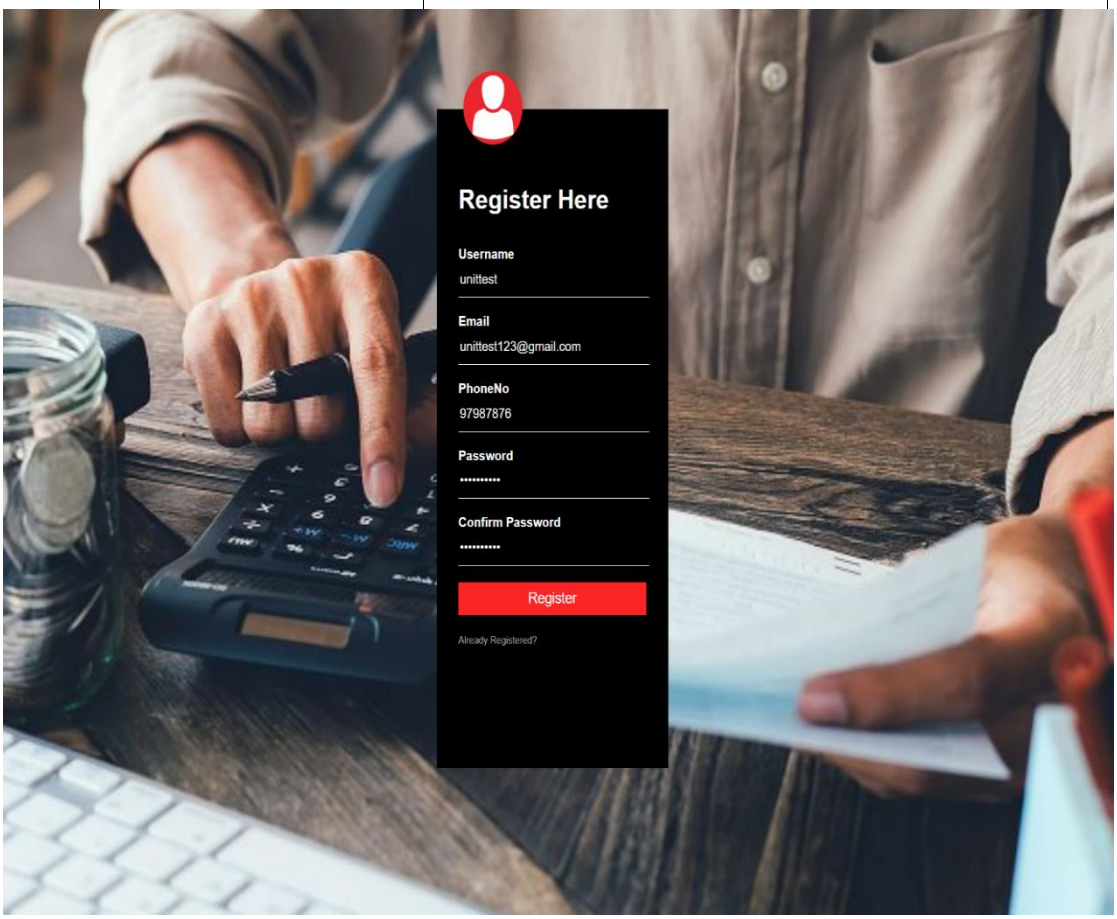

Figure 35: Unit testing registration Page

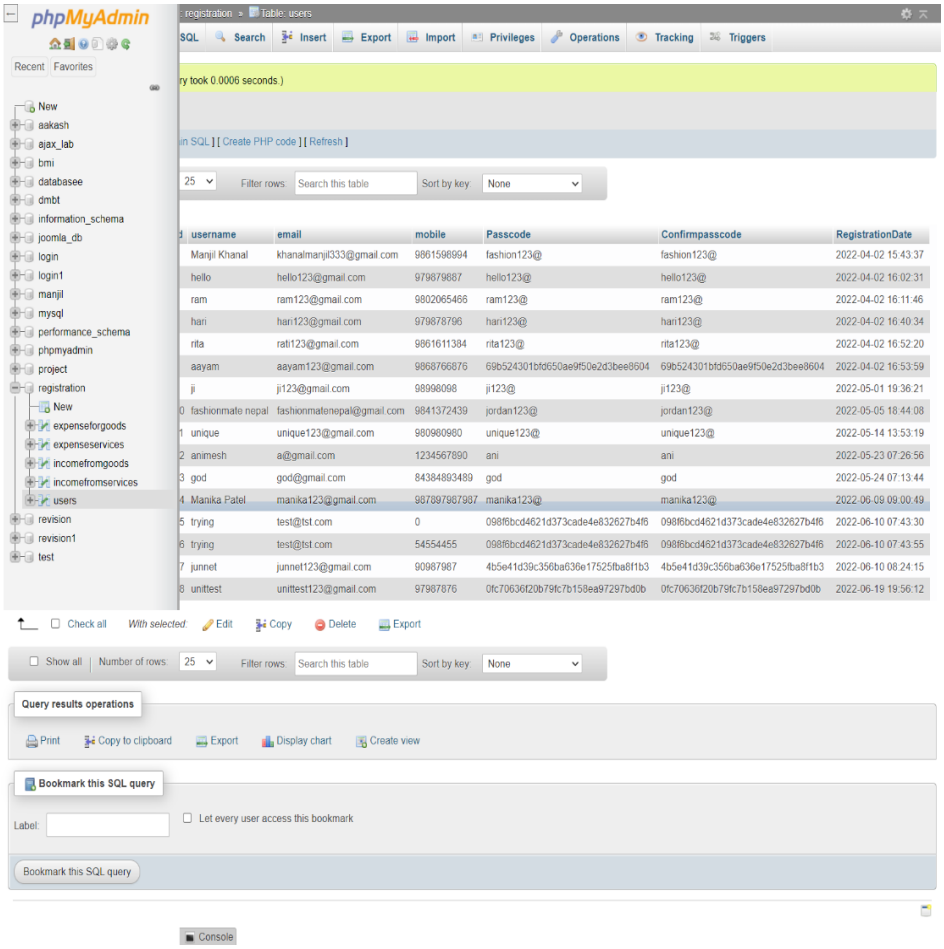

Figure 36: Unit Testing Registration (user account record)

User Page Login.

Objective	To login user account
Action	Fill the information required in the form and click login
Expected Results	Redirected directly to Dashboard page after successfully logging in the account by clicking on login button
Actual Results	Redirected to Dashboard page after successfully logging in the account by clicking on login button
Conclusion	Test Successful.

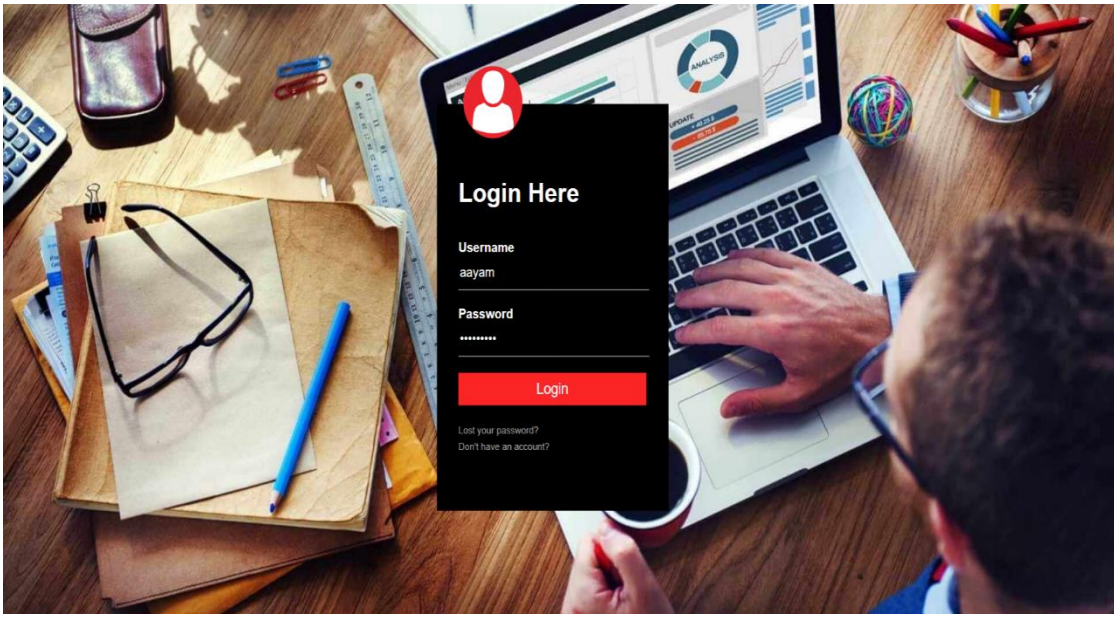

Figure 37: Unit Testing (Login Page)

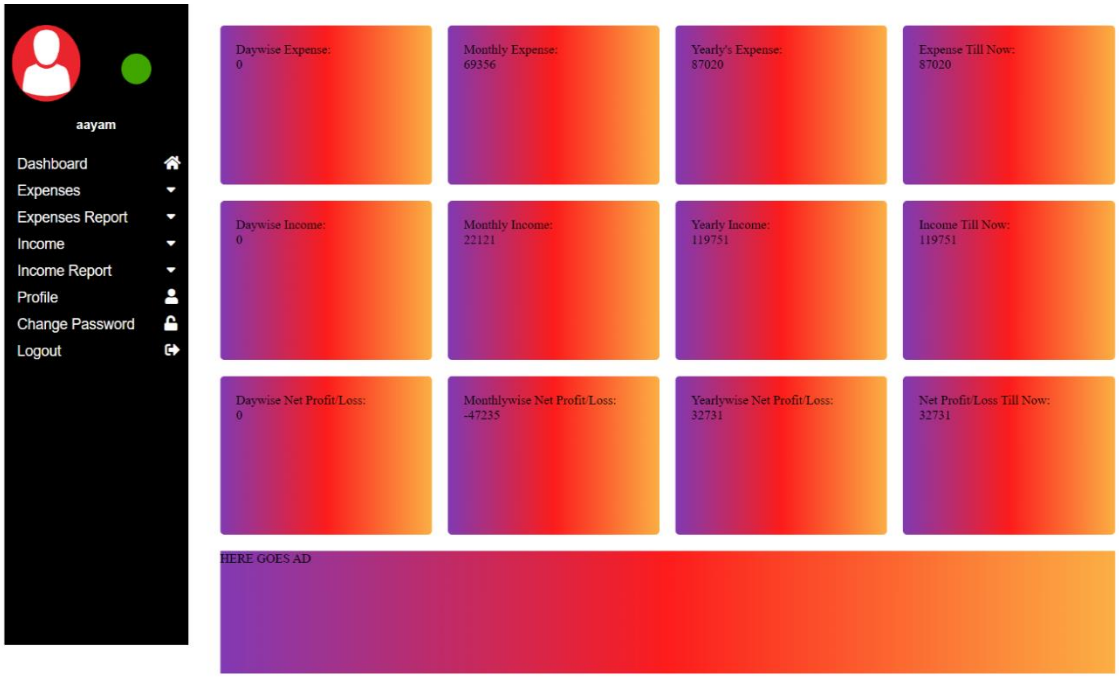

Figure 38: Unit Testing (Dashboard Page)

Add expense for services.

Objective	To add expense for services
Action	Fill the information required in the form and click submit
Expected Results	Successfully recording the records in the database.
Actual Results	Successfully recorded the records in the database.
Conclusion	Test Successful.

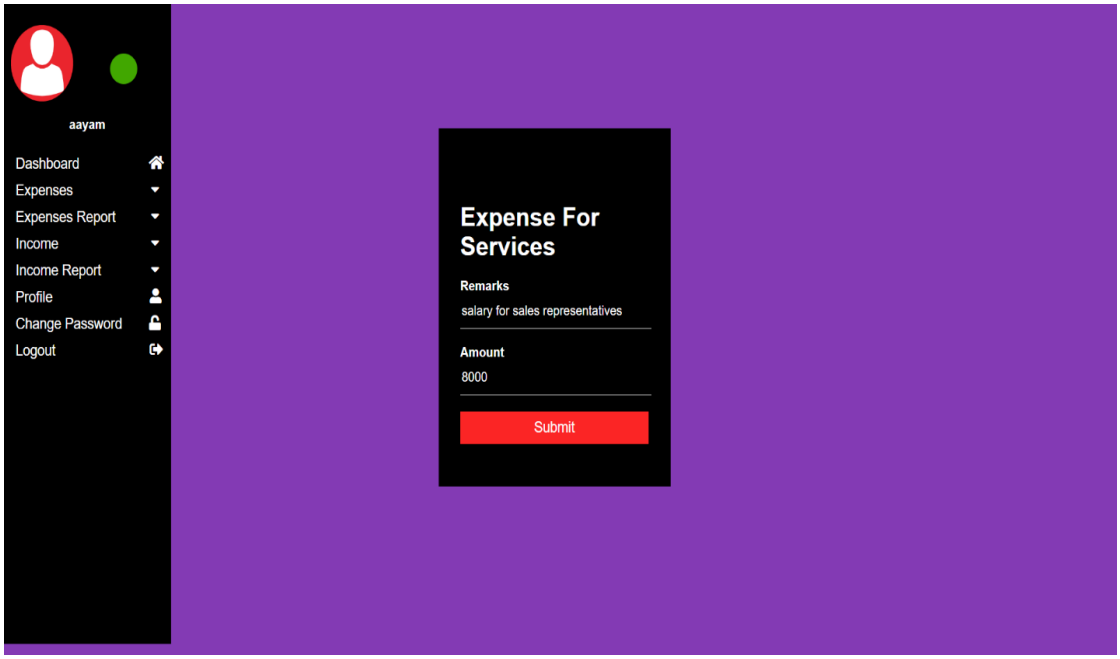

Figure 39: Unit testing for expense from services

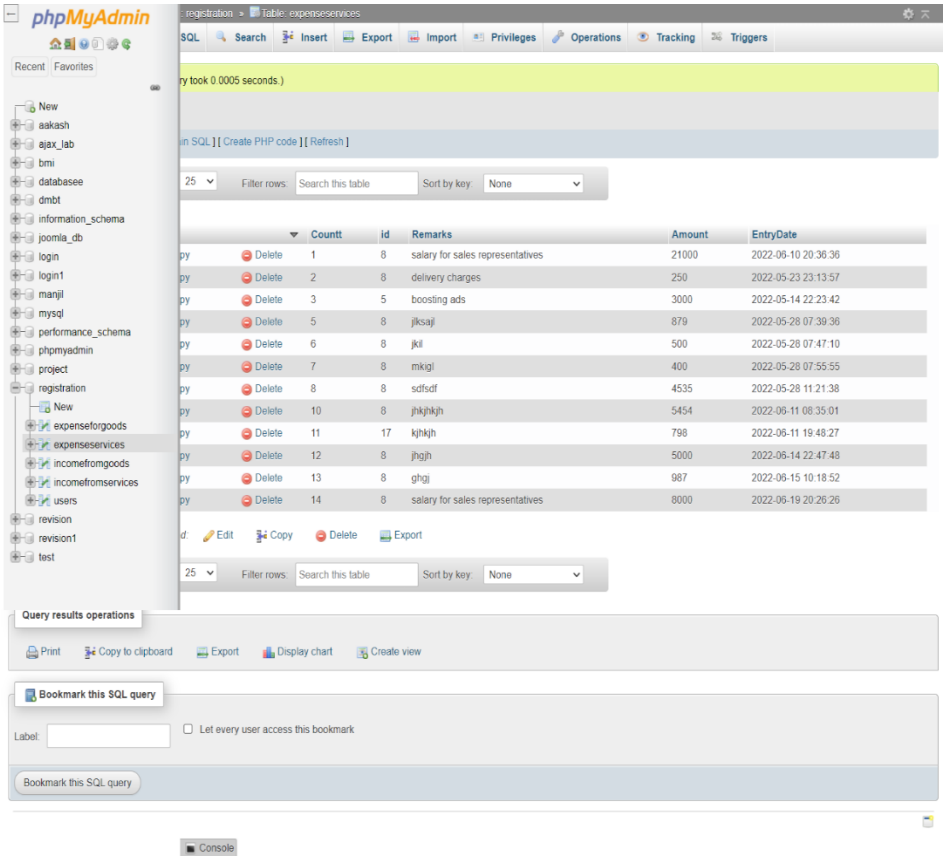

Figure 40: Unit testing database of expense from services

Add expense for goods.

Objective	To add expense for goods.
Action	Fill the information required in the form and click submit
Expected Results	Successfully recording the records in the database.
Actual Results	Successfully recorded the records in the database.
Conclusion	Test Successful.

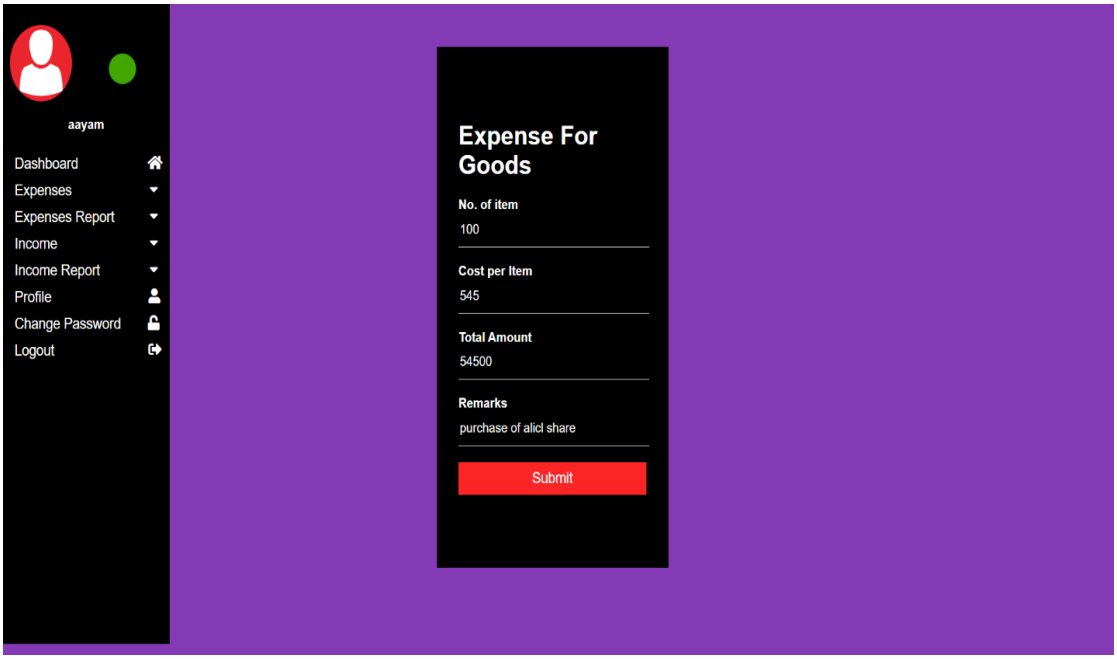

Figure 41: Unit Testing of Expense for Goods

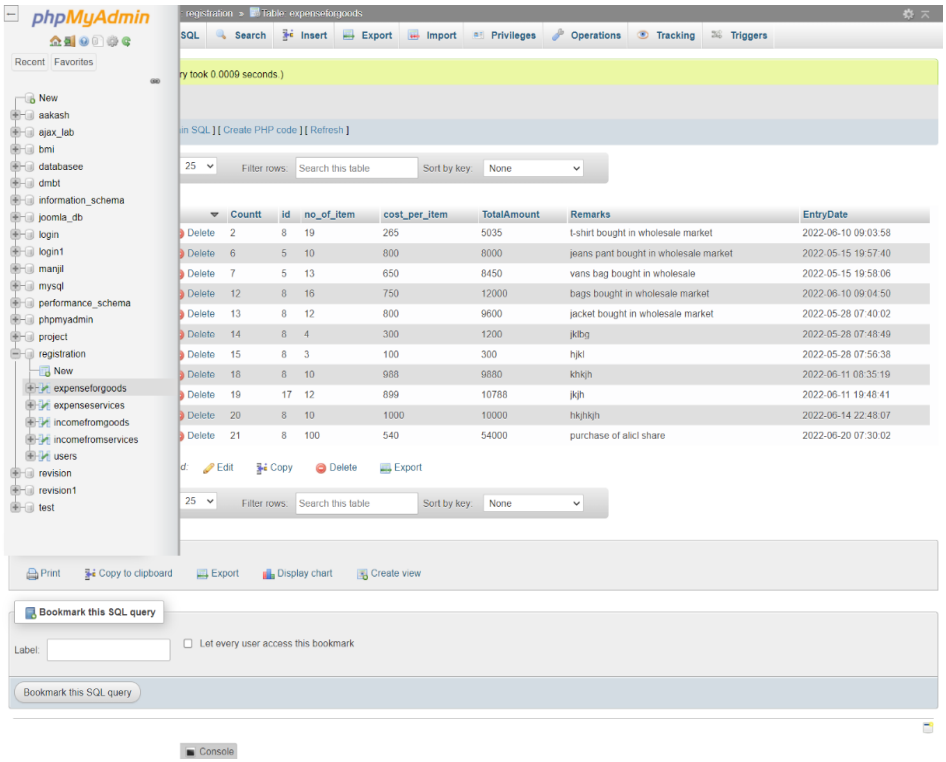

Figure 42: Unit testing for records of Expenses for Goods

Manage Expenses

Objective	To retrieve records of expense of goods and expense from services after selecting from date and to date.
Action	Select date required in the form and click submit
Expected Results	To successfully retrieve the records from the database where users can delete and update data from the selected date.
Actual Results	Successfully retrieved the records from the database where users can delete and update data but the select date is not working in which when we keep future date then also all data is being retrieved.
Conclusion	Test Failed. Select date is not working and all data is being displayed.

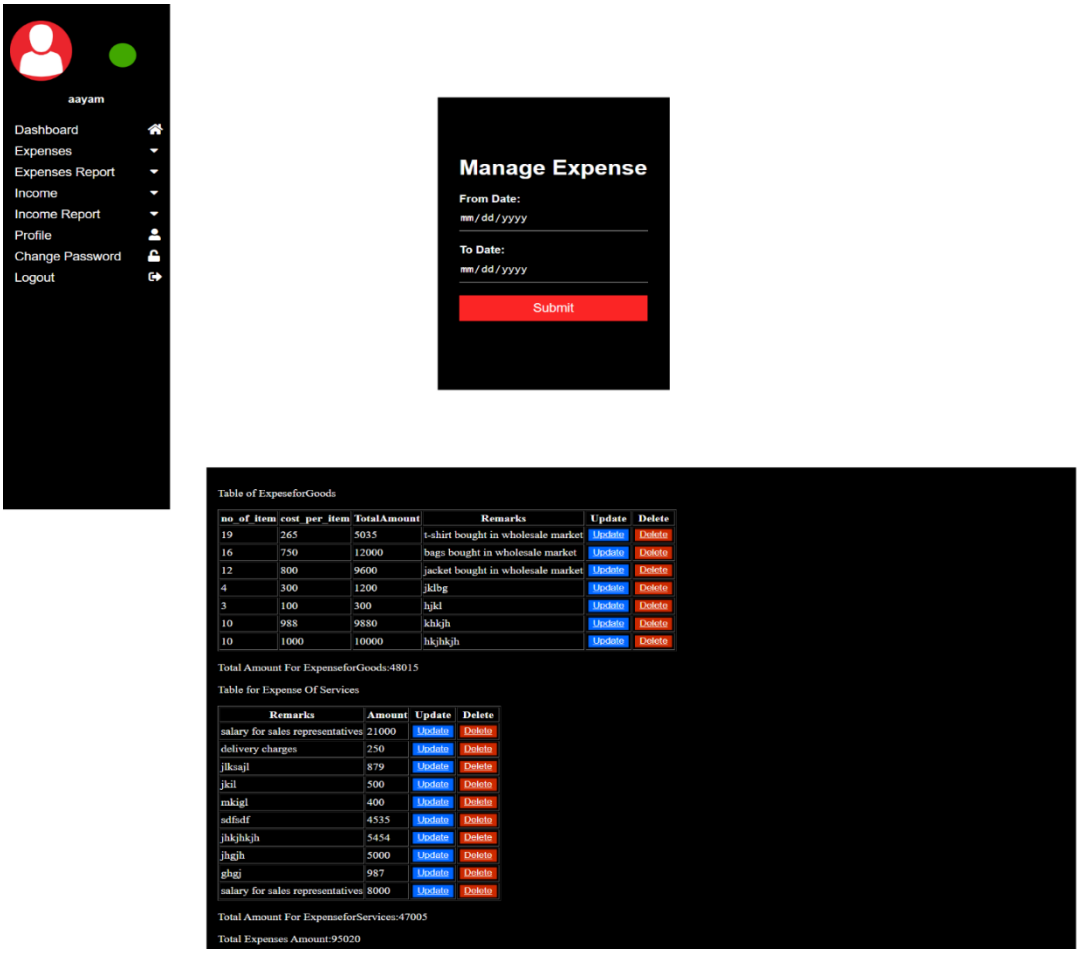

Figure 43: Unit Testing for Manage Expense

Manage Expenses

Objective	To retrieve records of expense of goods and expense from services after selecting from date and to date.
Action	Select date required in the form and click submit
Expected Results	To successfully retrieve the records from the database where users can delete and update data.
Actual Results	Successfully retrieved the records from the database where users can delete and update data.
Conclusion	Test Successful.

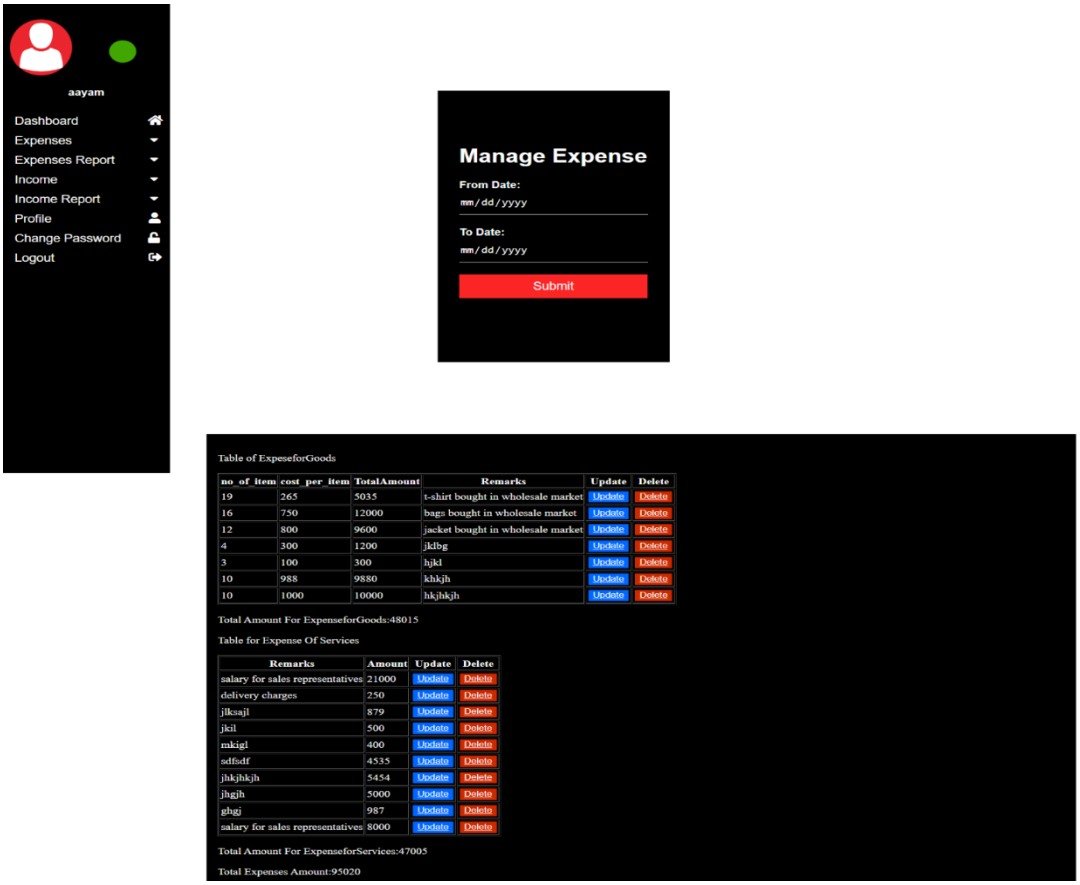

Figure 44: Unit testing for manage expense

For updating table of expense for goods

Objective	To update records of table of expense for goods
Action	Fill the information that needs to be changed in the form and click submit button.
Expected Results	The records in the table of expense for goods should be updated and the page should directly redirect to manage expense page.
Actual Results	The records in the table of expense for goods is updated and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

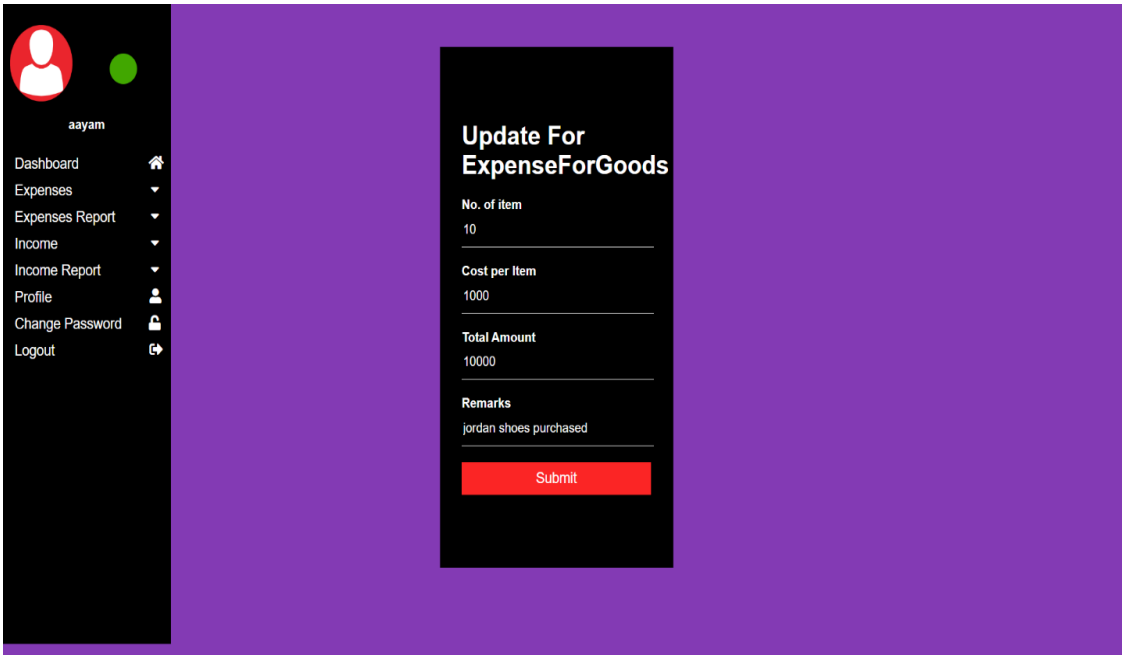

Figure 45: Unit testing for updating expense for goods

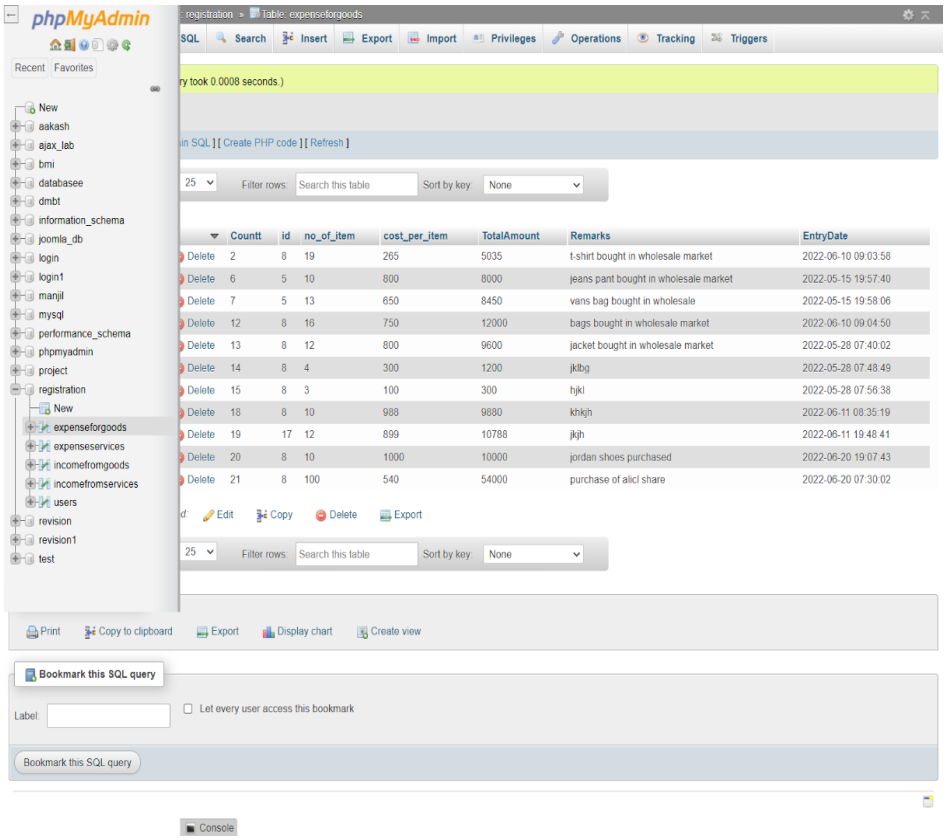

Figure 46: Unit testing for updated records of expense for goods

For updating table of expense from services

Objective	To update records of table of expense from services.
Action	Fill the information that needs to be changed in the form and click submit button.
Expected Results	The records in the table of expense from services should be updated and the page should directly redirect to manage expense page.
Actual Results	The records in the table of expense from services is updated and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

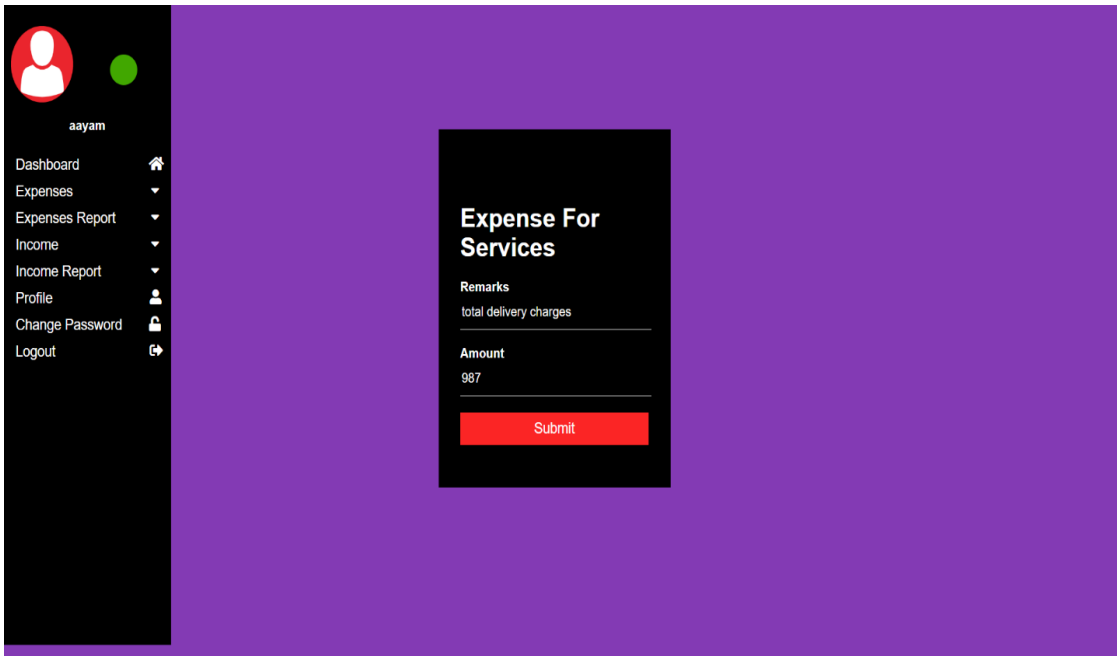

Figure 47: Unit testing for expense from services

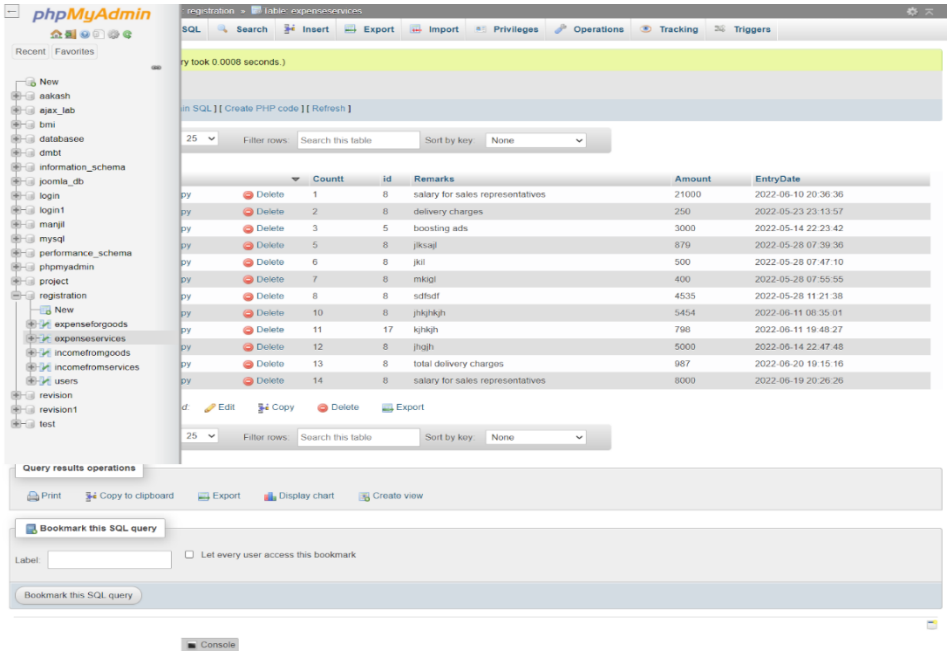

Figure 48: Unit testing (updated records of expense from services)

For deleting table of expense from services

Objective	To delete records of table of expense from services.
Action	Click on delete button on side of update button to delete the certain record in the table of expense from services.
Expected Results	The records in the table of expense from services should be deleted and the page should directly redirect to manage expense page.
Actual Results	The records in the table of expense from services is deleted and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

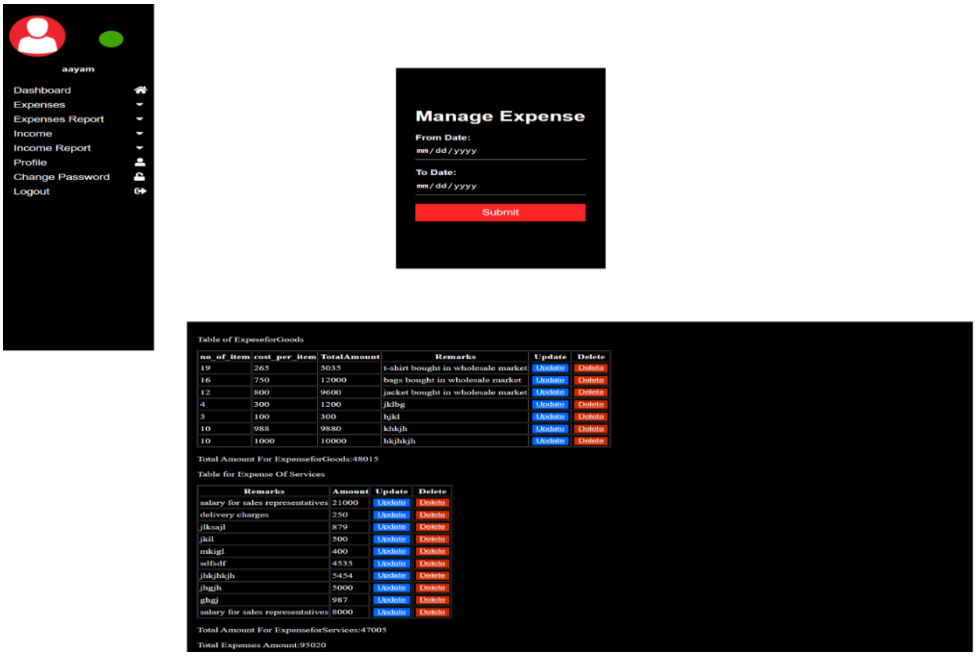

Figure 49: Unit testing for deleting expense from services before deleting

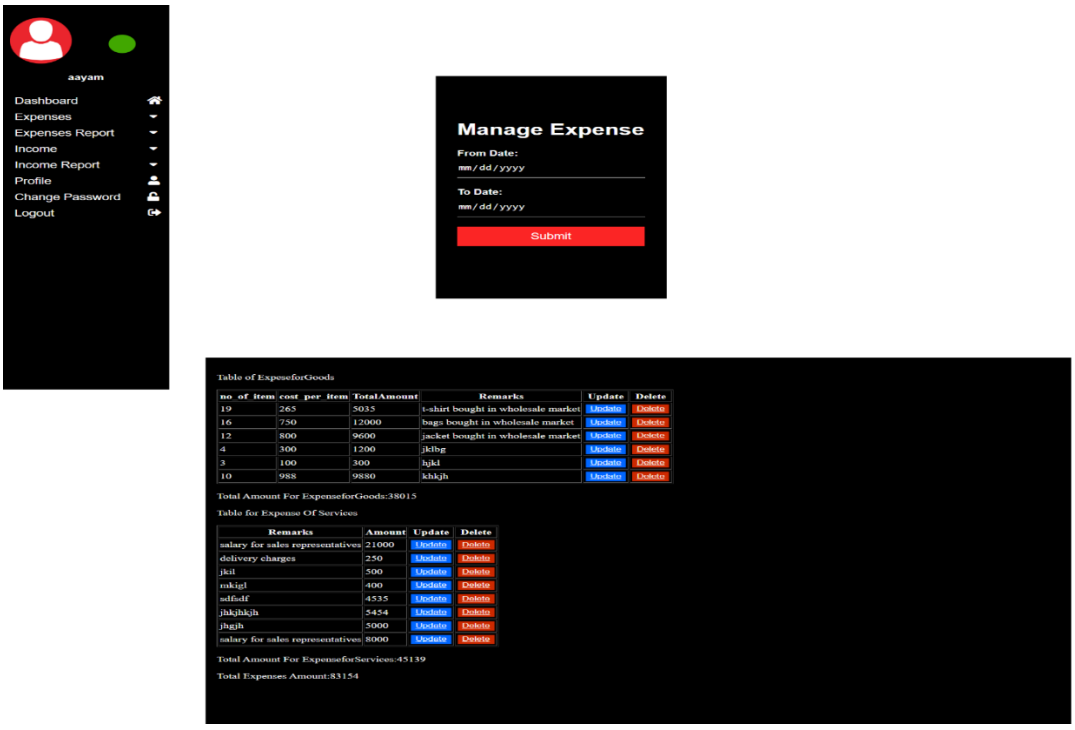

Figure 50: Unit testing after deleting records of expenses from services

For deleting table of expense for goods

Objective	To delete records of table of expense for goods.
Action	Click on delete button on side of update button to delete the certain record in the table of expense for goods.
Expected Results	The records in the table of expense for goods should be deleted and the page should directly redirect to manage expense page.
Actual Results	The records in the table of expense for goods is deleted and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

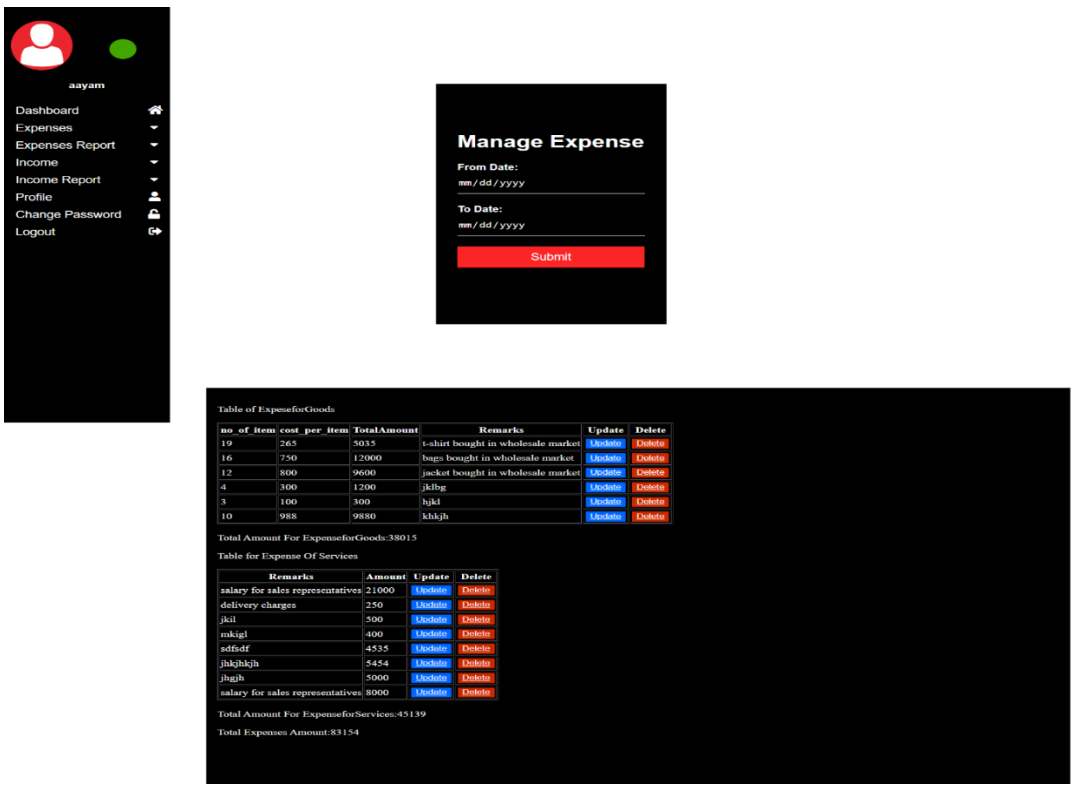

Figure 51: Unit testing before deleting record of expense for goods

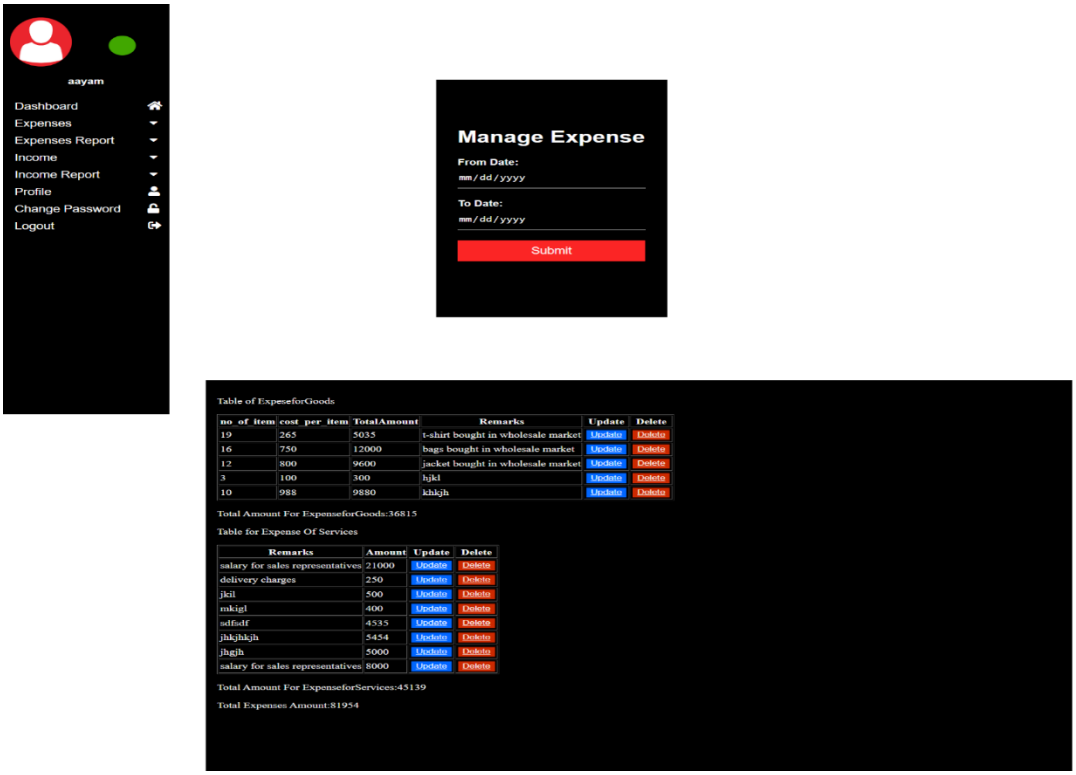

Figure 52: Unit testing after deleting records of expense for goods

Add income from services.

Objective	To add income from services
Action	Fill the information required in the form and click submit
Expected Results	Successfully recording the records in the database.
Actual Results	Successfully recorded the records in the database.
Conclusion	Test Successful.

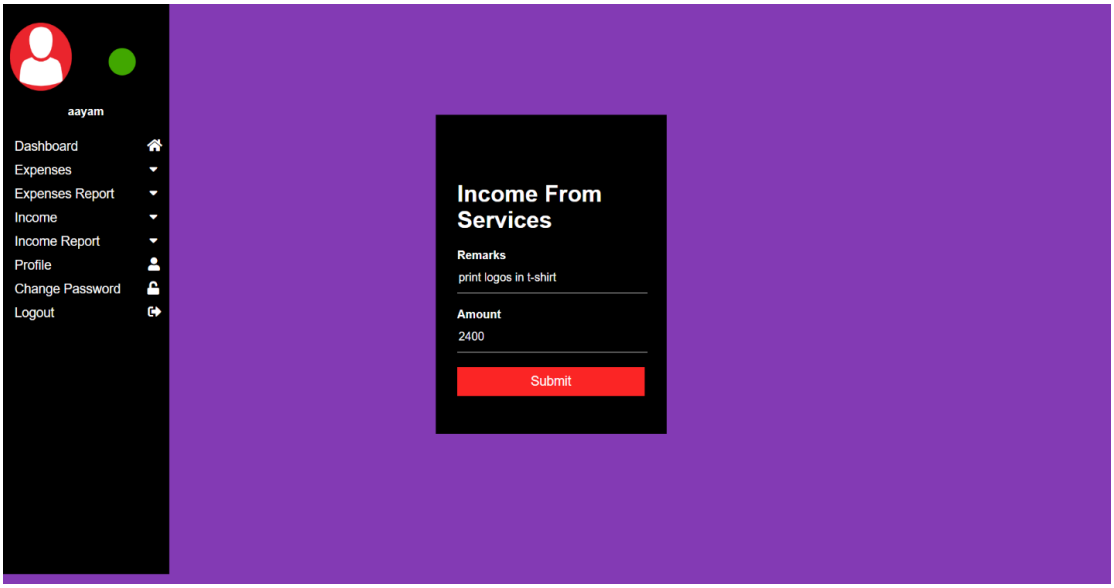

Figure 53: Unit testing for adding income from services

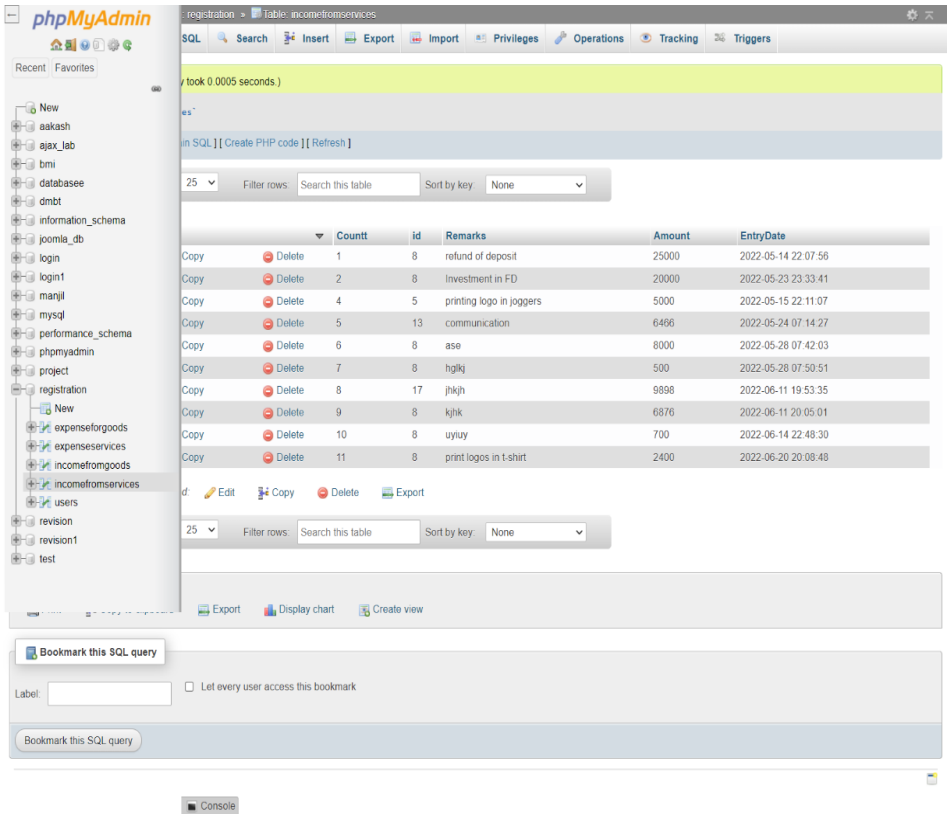

Figure 54: Unit testing for adding income from services records

Add income for goods.

Objective	To add income for goods.
Action	Fill the information required in the form and click submit
Expected Results	Successfully recording the records in the database.
Actual Results	Successfully recorded the records in the database.
Conclusion	Test Successful.

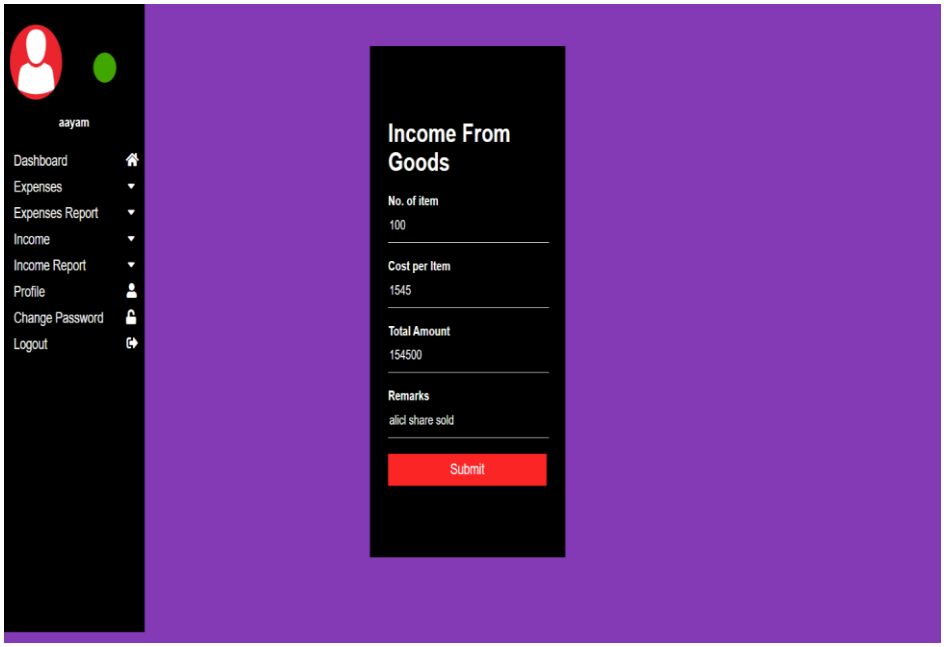

Figure 55: Unit testing for adding income from goods

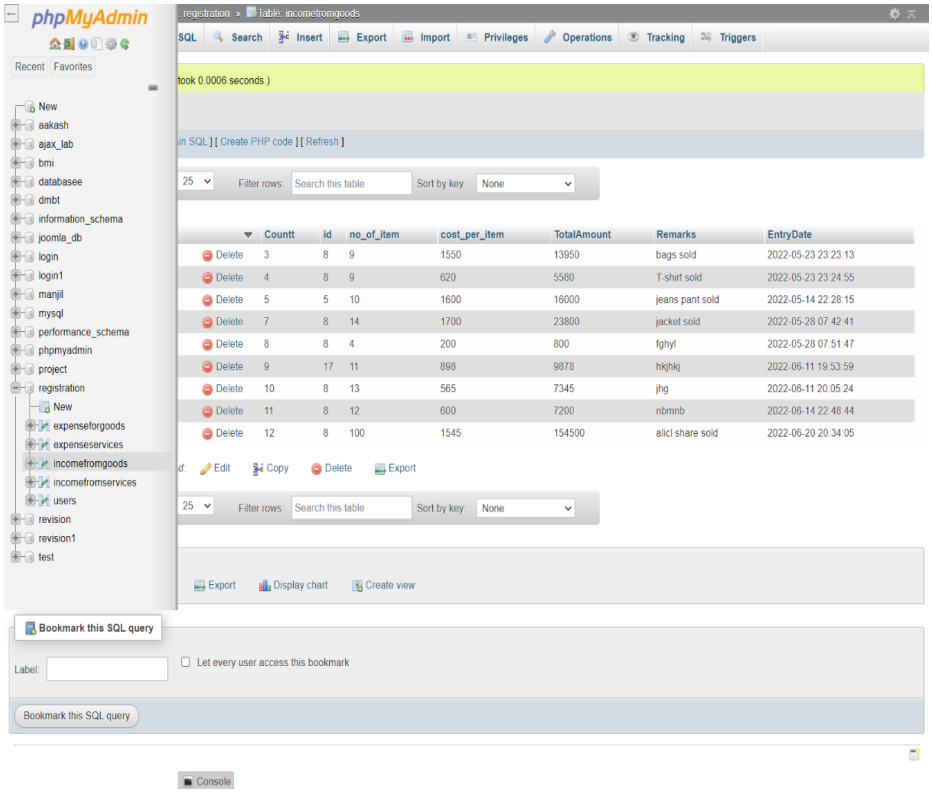

Figure 56: Unit testing for adding income from goods records in database

Manage Income

Objective	To retrieve records of income from goods and income from services after selecting from date and to date.
Action	Select date required in the form and click submit
Expected Results	To successfully retrieve the records from the database where users can delete and update data.
Actual Results	Successfully retrieved the records from the database where users can delete and update data.
Conclusion	Test Successful.

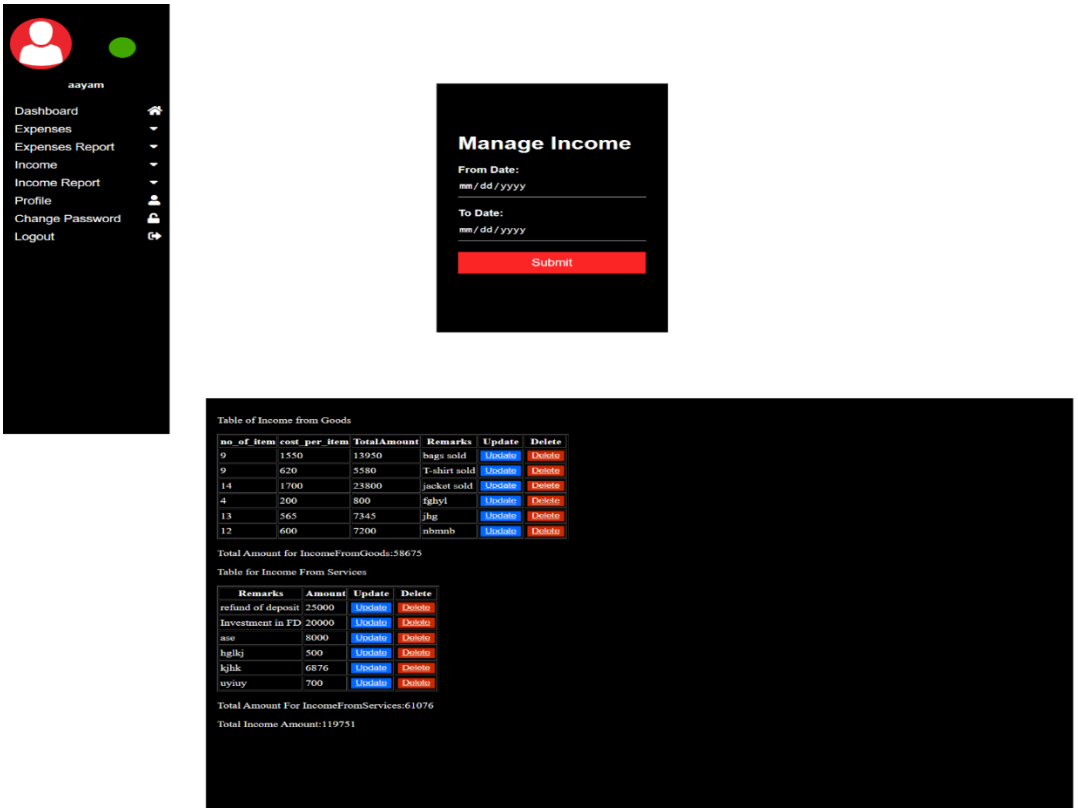

Figure 57: Unit testing for manage income

For updating table of Income from Goods

Objective	To update records of table of income from goods
Action	Fill the information that needs to be changed in the form and click submit button.
Expected Results	The records in the table of income from goods should be updated and the page should directly redirect to manage expense page.
Actual Results	The records in the table of income from goods is updated and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

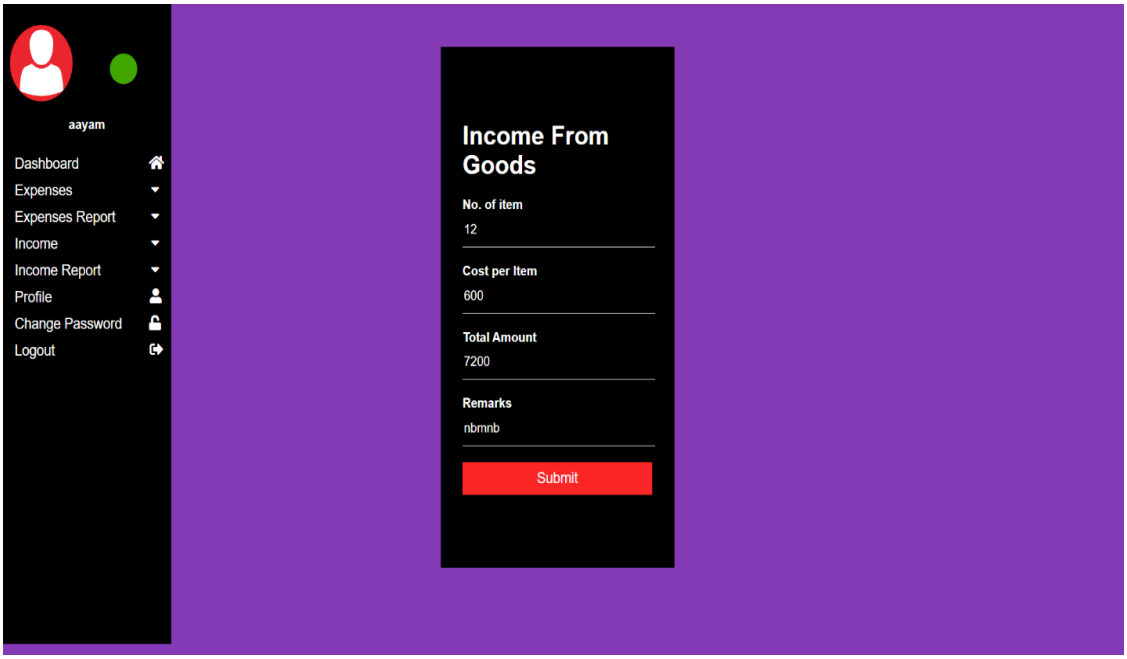

Figure 58: Unit testing for Updating income from goods

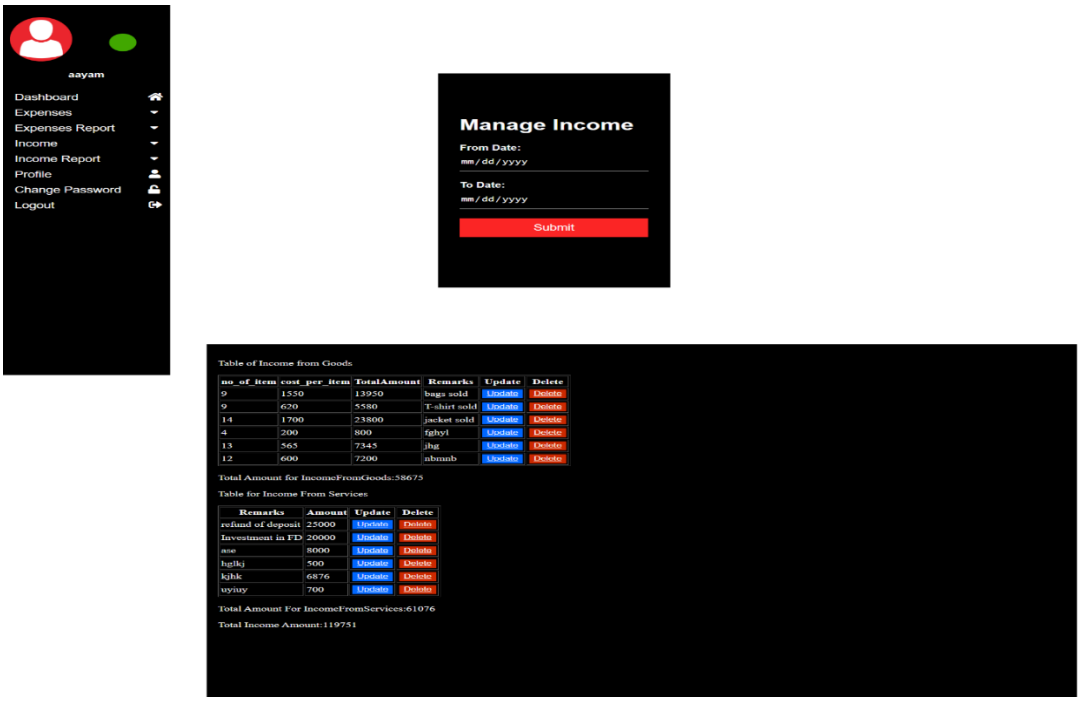

Figure 59: Unit testing Income from goods before updating

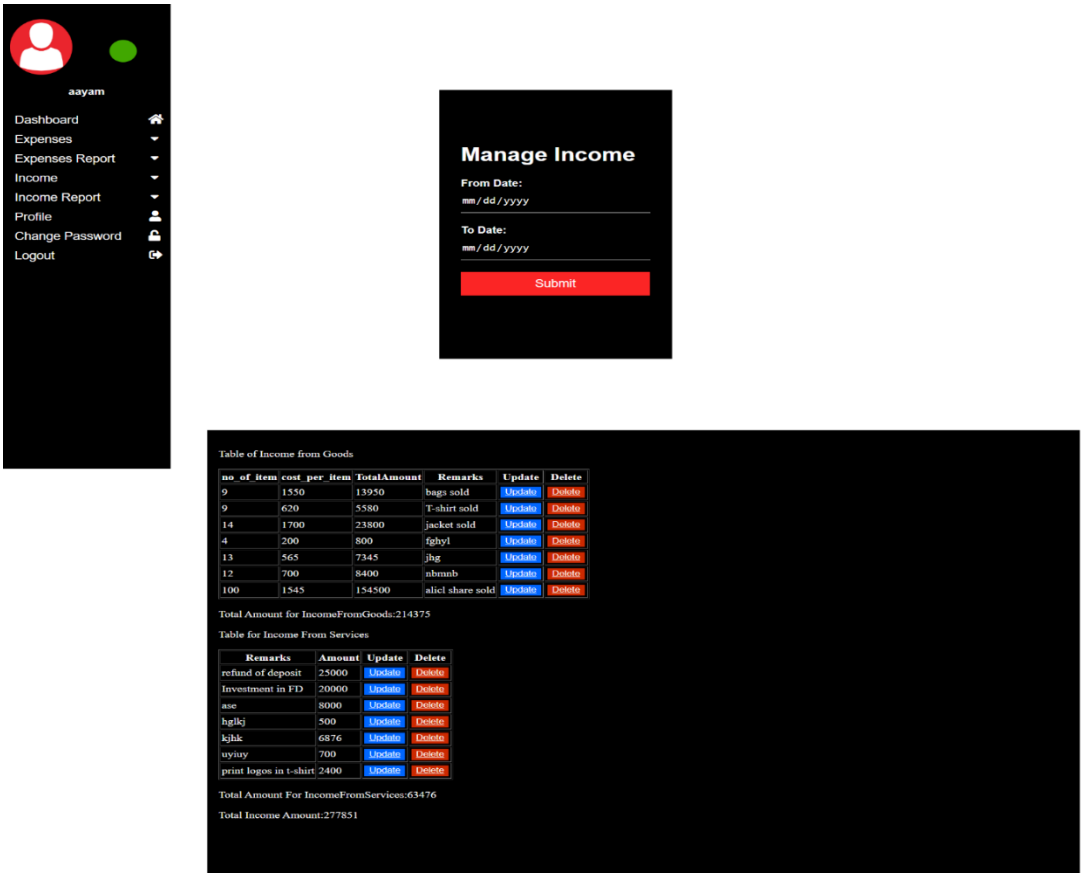

Figure 60: Unit testing Income from Goods after updating

For updating table of Income from Services

Objective	To update records of table of Income from services.
Action	Fill the information that needs to be changed in the form and click submit button.
Expected Results	The records in the table of Income from services should be updated and the page should directly redirect to manage expense page.
Actual Results	The records in the table of Income from services is updated and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

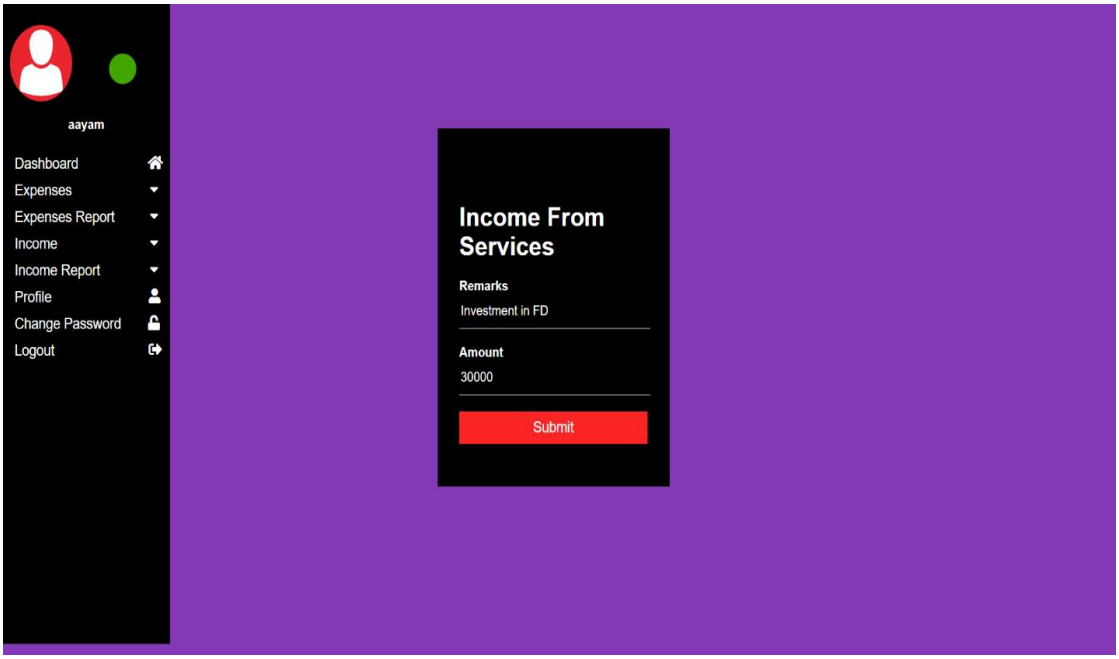

Figure 61: Unit testing for Updating income from services

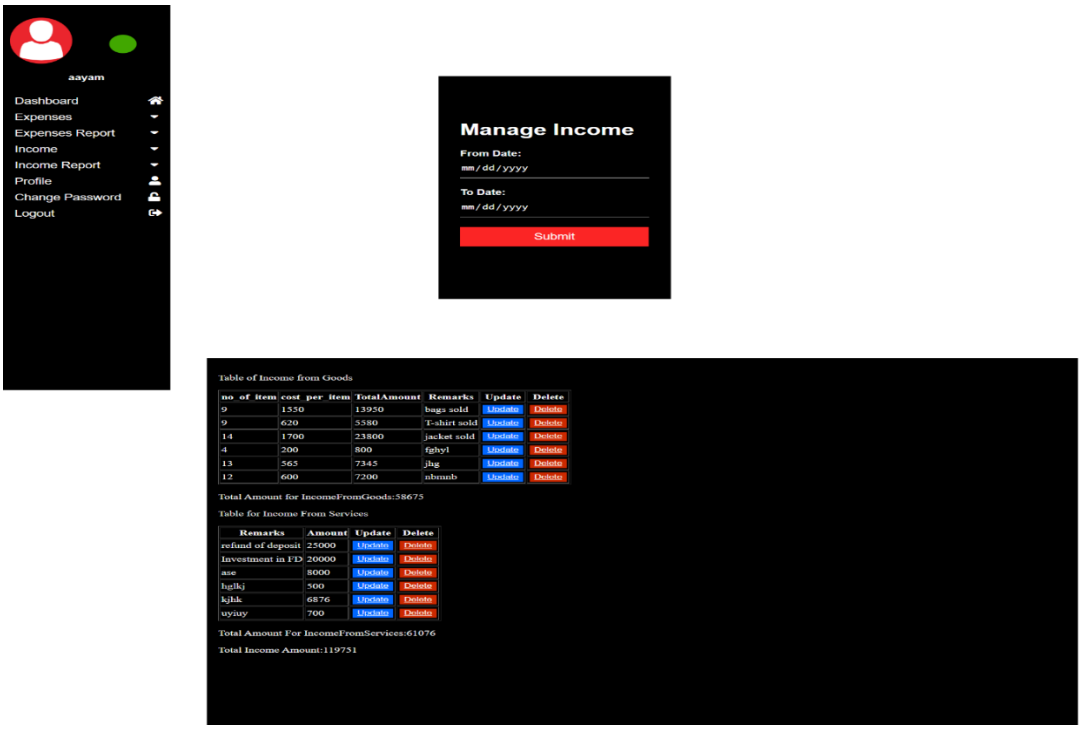

Figure 62: Unit testing for updating income from services before updating

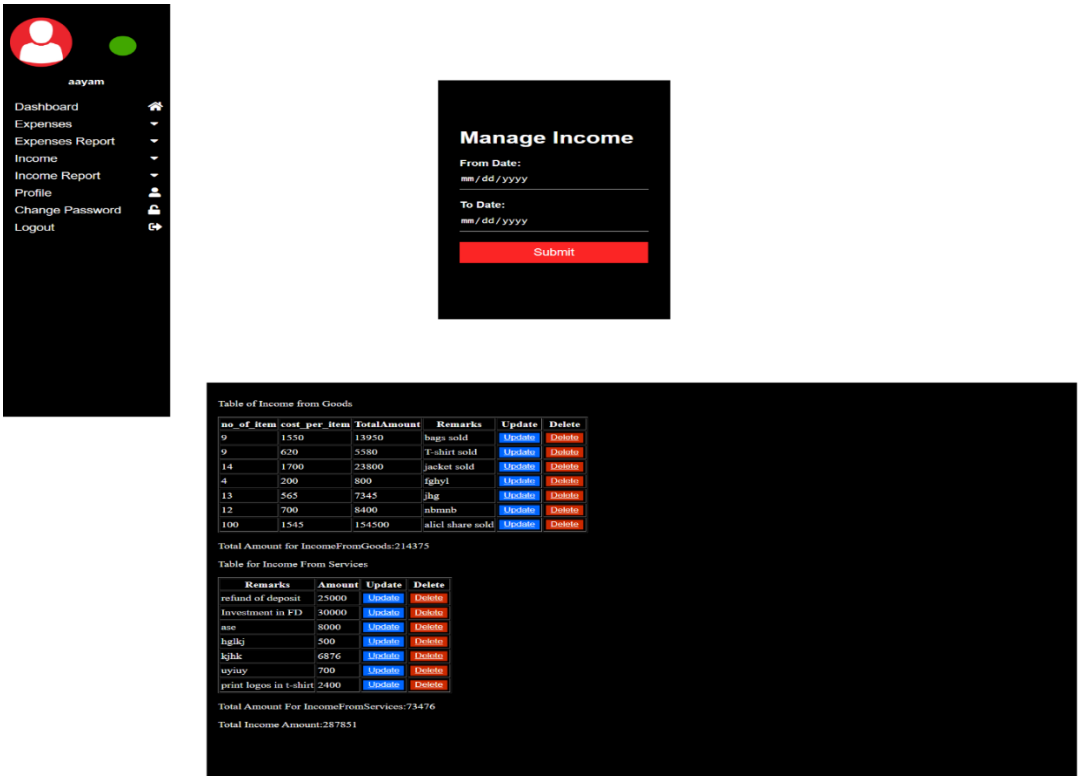

Figure 63: Unit testing for updating income from services after updating

For deleting table of income from services

Objective	To delete records of table of income from services.
Action	Click on delete button on side of update button to delete the certain record in the table of income from services.
Expected Results	The records in the table of income from services should be deleted and the page should directly redirect to manage expense page.
Actual Results	The records in the table of income from services is deleted and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

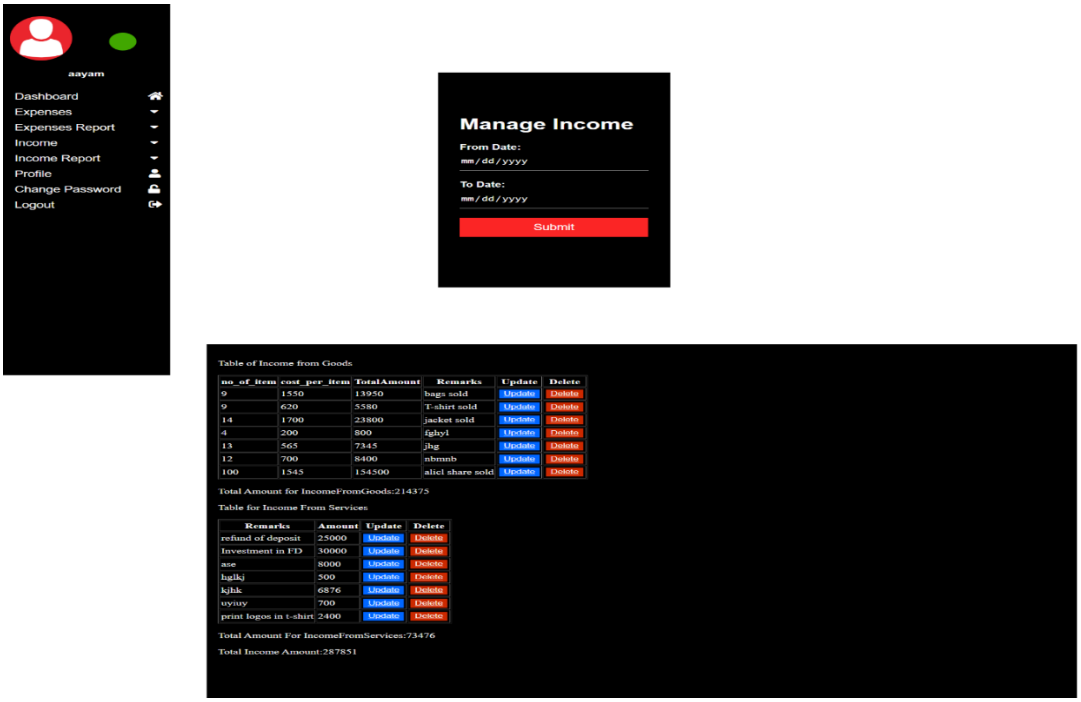

Figure 64: Unit testing before deleting records of income from services

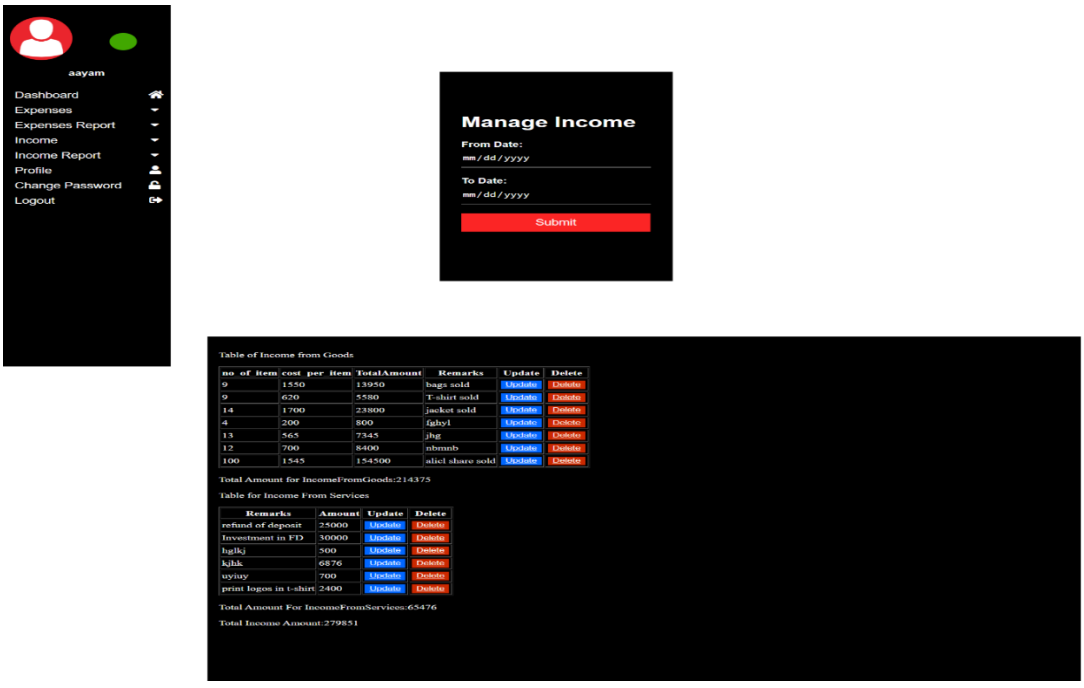

Figure 65: Unit testing after deleting record of income from services

For deleting table of income from goods

Objective	To delete records of table of income from goods.
Action	Click on delete button on side of update button to delete the certain record in the table of income from goods.
Expected Results	The records in the table of income from goods should be deleted and the page should directly redirect to manage expense page.
Actual Results	The records in the table of income from goods is deleted and the page is directly redirected to manage expense page.
Conclusion	Test Successful.

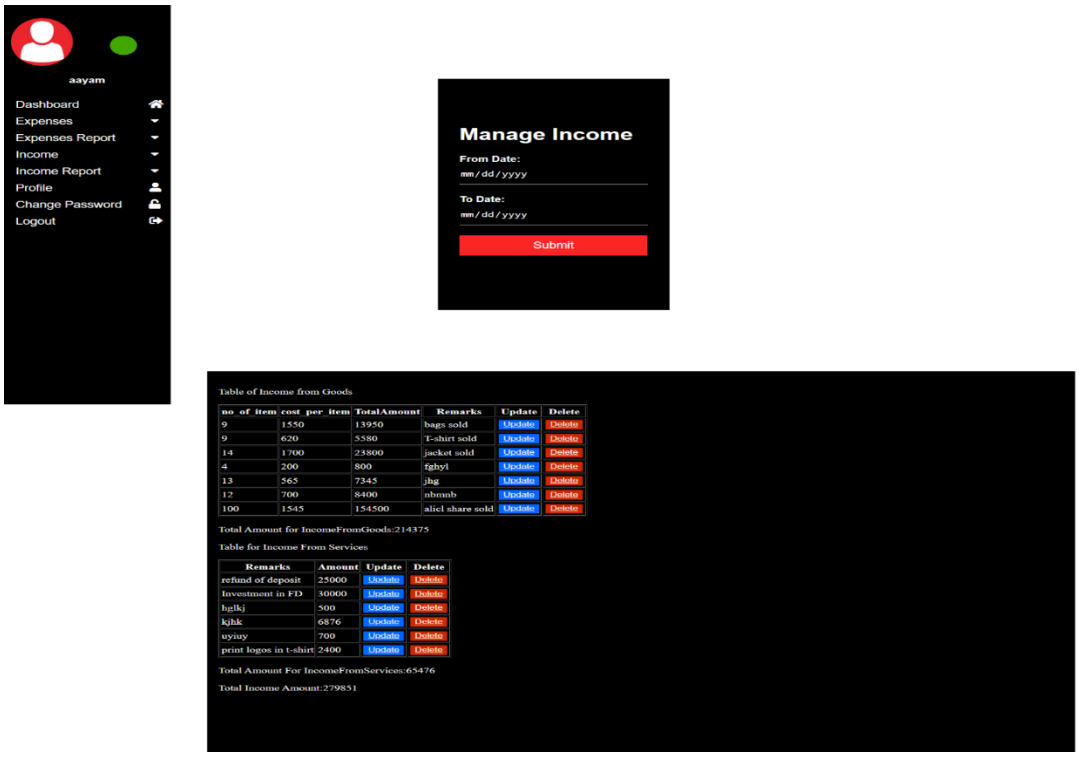

Figure 66: Unit testing for income from goods before deleting records

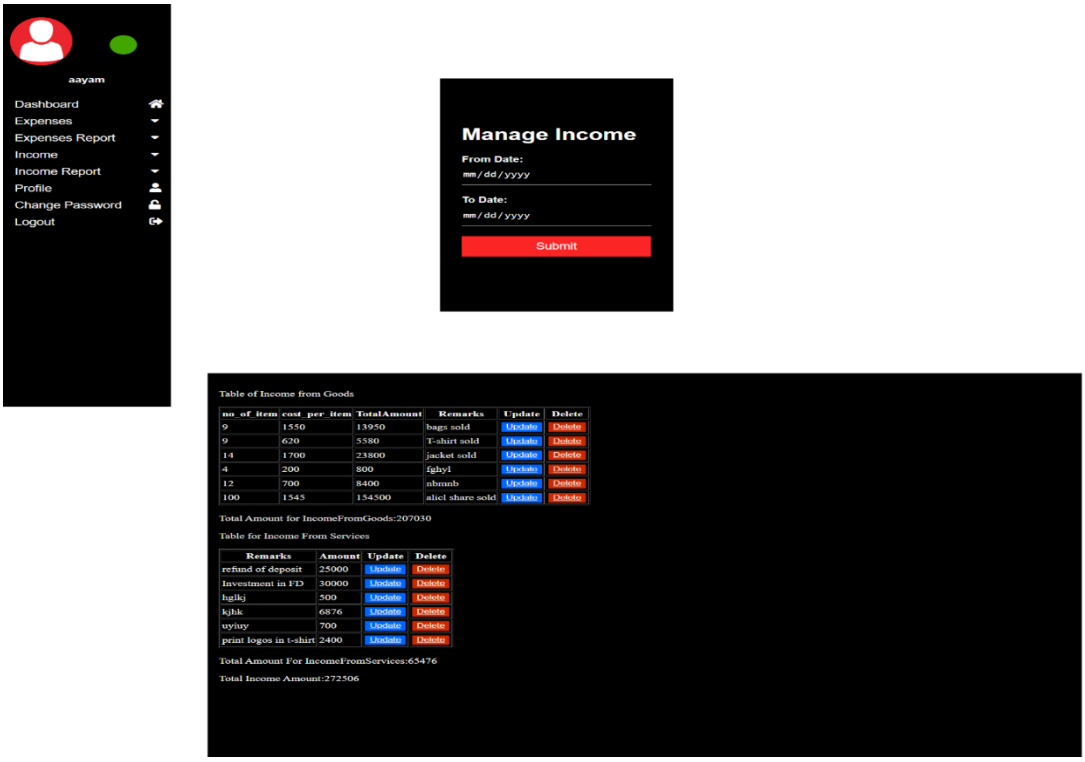

Figure 67: Unit testing after deleting records of income from goods

For updating user profile

Objective	To update user profile.
Action	Change the information in the form and submit the form and successful message will be displayed.
Expected Results	User information will be changed in users table after clicking on submit button in the form and successful message will be displayed in the form.
Actual Results	User information is changed in users table after clicking on submit button in the form and successful message is displayed in the form.
Conclusion	Test Successful.

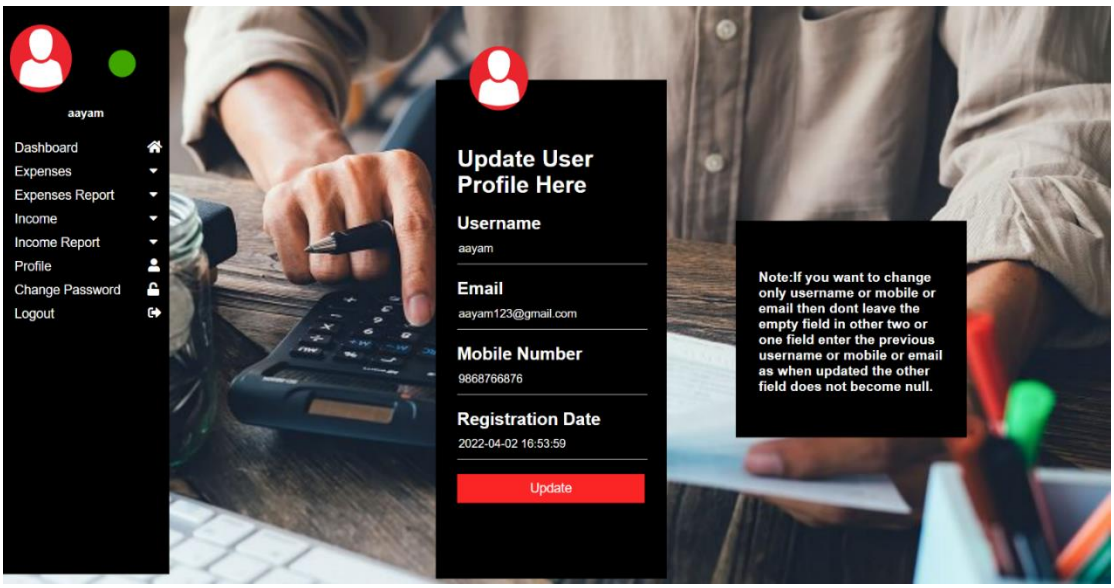

Figure 68: Unit testing before updating user profile

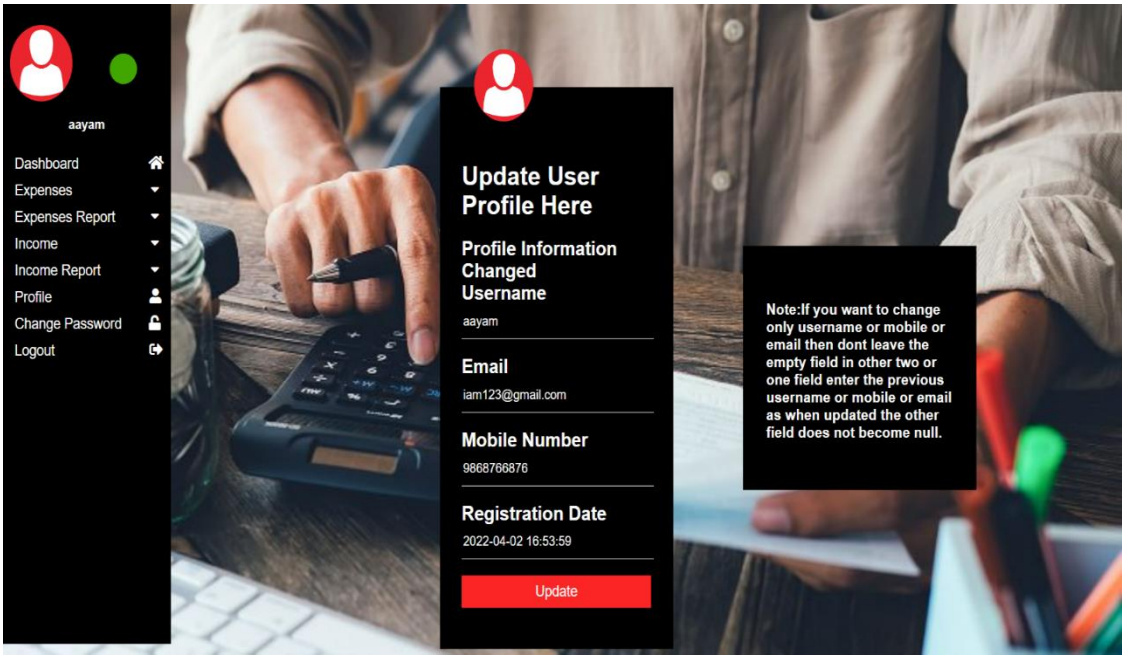

Figure 69: Unit testing after updating user profile

For updating password of user account

Objective	To update user account password.
Action	Change the information in the form and submit the form and successful message will be displayed.
Expected Results	User password will be changed in users table after clicking on submit button in the form and successful message will be displayed in the form.
Actual Results	User password is changed in users table after clicking on submit button in the form and successful message is displayed in the form.
Conclusion	Test Successful.

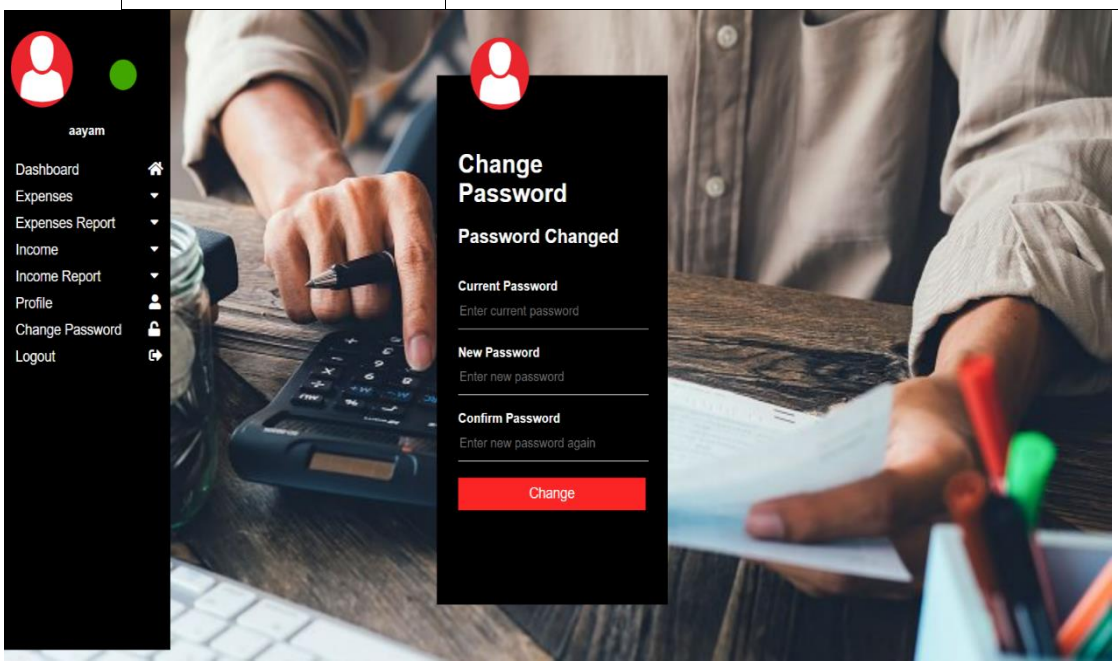

Figure 70: Unit testing for changing user account password

For forget password of user account

Objective	To send link for resetting the password.
Action	Fill the email address in the form and submit it.
Expected Results	User will get email for resting the password of user in which they can change the password.
Actual Results	User did not get the password reset link in the email address of user.
Conclusion	Test Failed.

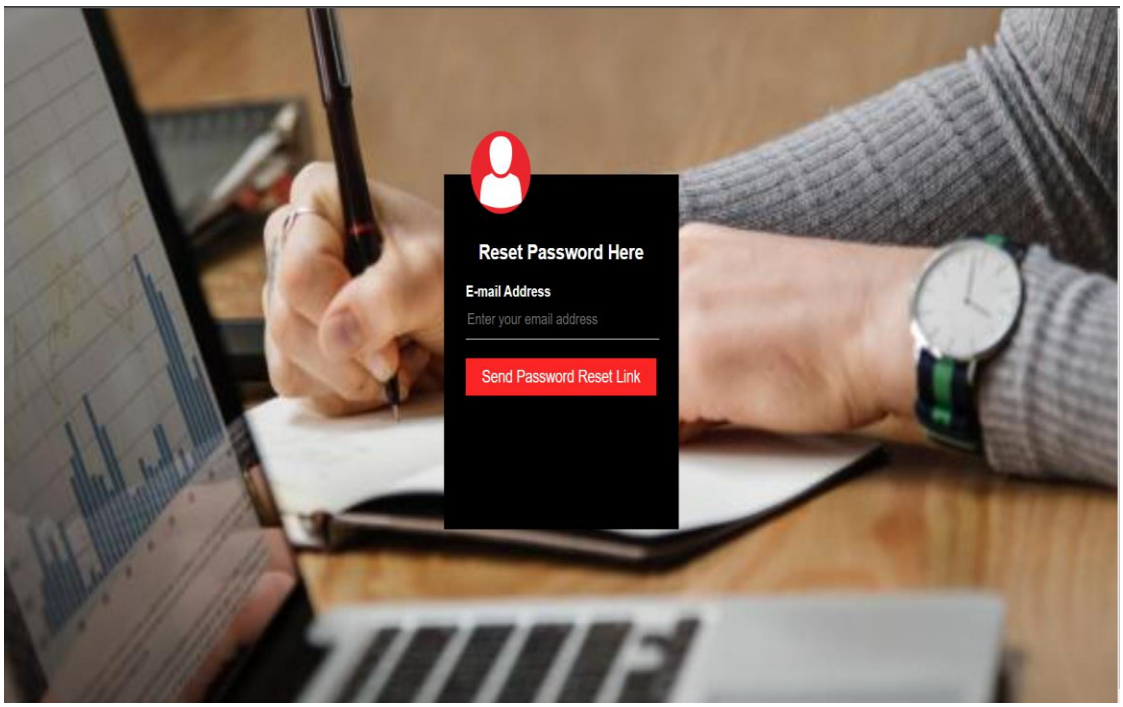

Figure 71: Unit testing for changing user account password

4.3.3 End to End testing

End-to-end testing is a Software testing methodology to test an application flow from start to end. The purpose of End-to-end testing is to simulate the real user scenario and validate the system under test and its components for integration and data integrity.

Nobody wants to be known for their mistakes and their negligence, and same is the case with the Testers. When the Testers are assigned an application to test, from that moment, they take the responsibility and the application also acts as a platform to show their practical and technical testing knowledge.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.2 Conclusion

In conclusion, this project is really useful at present because it focuses on business organization where day to day transactions is happening in which this system named "**Budget Tracker**" can track day to day transaction happening in the business organization from which customer can determine total loss and profit in the business organization.

Here, this system tracks daily expense for goods/services and income from goods/ services and determines total loss and profit and customer can view records that they have added in the system and can make changes in those system in which this system fulfills the use of CRUD operation. Here, in this system customer has facility to view records, total profit/loss in day-wise, monthly-wise, yearly-wise, and till now basis.

5.3 Future Recommendations

In the future, payment gateway will be added in this system for launching this system commercially through the use of API of E-sewa. And there is some part left in this system like for backend part of forget password page in which it will be done in future while lunching the system fully commercially. And other part of work left in this project is to add Google AdSense ads in the system from which extra revenue can be generated.

References

[1] PHP Gurukul, "Daily Expense Tracker using CodeIgniter," PHPGurukul, 2022.

[2] "SSLA," 2017. [Online]. Available: https://www.ssla.co.uk/spiral-sdlc/.

[3] SSLA, "Agile SDLC," *The difference between agile and waterfall*, p. 1, 2013.

[4] codewithawa, "Complete user registration system using PHP and MySQL database," *User registration system*, p. 1, 2013.

[5] thapatechnical, "peration in PHP MySQLi with Code. Select Insert Update Delete in PHP MySQLi," p. 2, 2018.

[6] Akunetix, "How to Prevent SQL Injection Vulnerabilities in PHP Applications," *SQL Injection in PHP*, 2022.

[7] M. A. K. Velupula, "Feasibility Study," *Studying the Importance and Feasibility of Software Testing ,* 2009.

[8] D. Doshi, "Spiral Model," *REVIEW OF THE SPIRAL MODEL AND ITS APPLICATIONS*, 2021.

[9] R. Koteeswaran, "SQL injection," *A study on SQL injection techniques*, 2016.

[10] C. Zhao, "What is SQL Injection?," *Research on SQL Injection Attack and Prevention Technology Based on Web*, 2019.

[11] A. Aleryani, "Comparative Study between Data Flow Diagram and Use Case Diagram," *Comparative Study between Data Flow Diagram and Use Case Diagram*, 2016.

[12] E. K. Park, "A Comparative Analysis of Entity-Relationship Diagrams," *A Comparative Analysis of Entity-Relationship Diagrams*, 1995.

[13] K. Froehlich, "Entity-relationship modeling," *Entity-relationship modeling*, 1994.

Appendix

FYP Logbook Entry Sheet

FYP Logbook Entry Sheet

Meeting No: 01 **Date: 2022/07/28**

Start Time: 10:00 am **End Time: 01:00 pm**

Items Discussed:
Font of the documentation, Use case Diagram and Contents of Documentation.

Achievements:

Font of documentation changed and Contents of Documentation modified.

Problems:

Problems faced for adding page number.

Tasks for Next Meeting: (Write down the task assigned to you by your supervisor)
To make changes in font of the documentation and to change the contents of the documentation.

Student Sign **Supervisor Sign**

FYP Logbook Entry Sheet

FYP Logbook Entry Sheet

Meeting No:02 **Date: 2022/07/29**

Start Time: 10:00 am **End Time: 01:00pm**

Items Discussed:

Verification done of process modeling and data modeling

Achievements:
Changes made in E-R diagram and format of pictures changed, copyright mark removed from the data modeling flow chart diagram.

Problems:
Copyright Mark in flowchart.

Tasks for Next Meeting: (Write down the task assigned to you by your supervisor)
To remove Copyright Mark in the Flowchart.

Student Sign **Supervisor Sign**

FYP Logbook Entry Sheet

FYP Logbook Entry Sheet

Meeting No:03 **Date: 2022/07/30**

Start Time: 10:00 am **End Time: 01:00 pm**

Items Discussed:
Chapter one introduction part verification for changes

Achievements:
Changes made in chapter one introduction part

Problems:
The main problem in introduction part was that the information in it was very short and it was explained in more elaborated way.

Tasks for Next Meeting: (Write down the task assigned to you by your supervisor)
To add more information in chapter one introduction part.

Student Sign **Supervisor Sign**