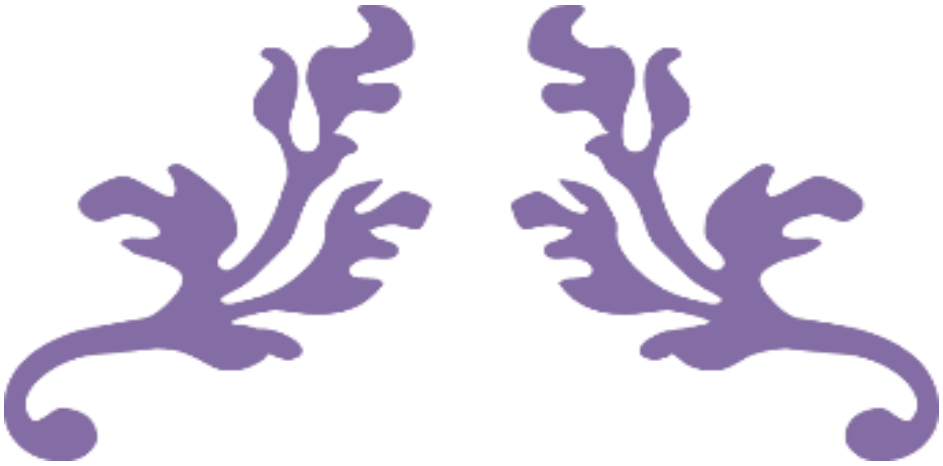


EXPENSER

Project Report

APRIL 5, 2024

VEER NARMAD SOUTH GUJARAT UNIVERSITY

ATMANAND SARASWATI SCIENCE COLLEGE

B.Sc. (Bachelor of Computer Science) Programme

PROJECT REPORT ON

EXPENSER

AS PARTIAL REQUIREMENT, FOR

B.Sc. (Computer Science) 6th Semester

Year: 2021 – 2024

GUIDED BY:

Mr. Jayesh Maurya

SUBMITTED BY:

Mr. Mahesh Bardoliya

(CS2103)

ACKOWLEDGMENT

With immense pleasure and a sense of fulfilment, We would like to present this report on the project entitled ‘Expenser.

We would like to express our sincere gratitude towards our H.O.D. Ms. Vaishnavi Jariwala & Project Guide Mr. Jayesh Maurya, for providing us the opportunity to work under her guidance. We are grateful to her for keeping faith in us and considering us worthy for such a development work. We thank her for being our torchbearer and for guiding us well to face the development challenges. We are thankful to her for taking personal interest and making all the resources available to us. Her valuable guidance and substantial suggestions were extremely helpful to us during the development of the project. We would also like to offer our gratitude towards our Principal Dr. Shailesh Padsala and all faculty members of our college, who helped us by giving their valuable suggestions and encouragement which not only helped us in preparing this report but also in having a better insight in this field. Overall, this project has given me a rich set of experience, which shall always guide me throughout my career.

Last but not the least; We owe a special debt to the All mighty for his grace and blessings. We would like to express our heartiest gratitude to our family for their love, faith and blessings which they bestowed us.

Thanking you all,

Mahesh Bardoliya

COLLEGE CERTIFICATE

ATMANAND SARASWATI SCIENCE COLLEGE

Managed By: Shree Tapi Brahmcharyashram Sabha, Surat.
Shree Swami Atmanand Saraswati Vidya Sankul, Kapodra,
Varachha Road, Surat-395006.

CERTIFICATE

This document is to certify that,

The Project entitled _____ has been carried out by _____

_____

Studying in T.Y.B.Sc. (Computer Science) _____ Semester for Academic year 20 - 20 .

Roll No: _____

Exam Seat No: _____

Project Guide **Head of Department** **External Examiner**

ABSTRACT

Expenser is a web application developed using the Django framework, leveraging PostgreSQL as its backend database management system. It is designed to offer users a comprehensive platform for managing expenses and income sources efficiently. The application provides a seamless experience for creating accounts, adding, removing, editing, and searching for expenses and income sources.

Expenser boasts an intuitive user interface designed to enhance user experience, featuring easy navigation and accessibility to various functionalities. Moreover, the application integrates advanced data analysis capabilities, enabling users to gain insights into their financial activities through interactive charts and graphs.

Following the Model-View-Controller (MVC) architecture pattern, Expenser ensures a clear separation of concerns, facilitating maintainability and scalability. The implementation emphasizes clean code practices and harnesses the power of Django's Object-Relational Mapping (ORM) for efficient database management with PostgreSQL.

Through rigorous testing methodologies, Expenser ensures reliability and robustness, while also providing a roadmap for future enhancements and optimizations. Overall, Expenser serves as a versatile tool for individuals seeking to streamline their financial management processes effectively.

INDEX

INTRODUCTION

1.1 Project Profile

Project Name:	Expenser
Technology:	Django
Front End:	HTML,CSS,JS
Back End:	PostgreSQL
Development Tool:	1.87.1
Documentation Tool:	Microsoft Word
Project Duration:	JANUARY 1 2024 TO APRIL 5 2024
Developed By:	Mahesh Bardoliya CS2103

Environment Description

2.1 Hardware Requirement

❖Development time Hardware Requirements

Microprocessor:	Intel(R) Core(TM) i5
Front End:	8.00 GB RAM
Back End:	1 GB

❖Runtime Hardware Requirements

Memory:	4 GB Minimum

2.2 Software Used

❖Development time Software Requirement

Operating System:	Windows 11
Server-Side Script Language:	Python
Database:	PostgreSQL
Browser:	Google Chrome
Server:	WSGI server
Software:	Visual Studio Code

Existing Planning

3.1 Manual process

❖ **User Registration and Login**: Users need to manually register for an account by providing necessary information such as username, email, and password. After registration, users need to log in with their credentials to access the application's features.

❖ **Expense Management**: Users manually input details such as expense amount, category, and description. Users can manually delete expenses from their account. Users can manually edit expense details like amount, category, or description.

❖ **Income Management**: Users manually input details such as income amount, source, and description. Users can manually delete income sources from their account. Users can manually edit income source details like amount, source, or description.

❖ **Email Verification**: Users need to manually check their email inbox for the verification link sent by Expenser after registration. Users click on the verification link manually to confirm their email address and activate their account.

❖**User Interface**: Users navigate through different sections of the application manually by clicking on links/buttons provided in the user interface. Users manually interact with elements on the interface to perform actions such as adding expenses, removing income sources, etc.

❖**Analytical Tools**: Users manually access the analytics section to view charts representing their expenses and income over time. Users manually interpret the data represented in the charts to understand their financial patterns and trends.

3.2 Manual process drawbacks

❖**Human Error**: Manual data entry increases the risk of human error. Users may input incorrect expense/income details, leading to inaccurate financial records.

❖**Time-consuming**: Manual processes, such as adding, updating, or removing expenses/income sources, can be time-consuming, especially for users with large amounts of data to manage.

❖**Inconsistency**: Users may not consistently categorize expenses or income sources, leading to inconsistencies in data organization. This inconsistency can hinder the accuracy of financial analysis and reporting.

❖**Limited Scalability**: As the number of users or the volume of data increases, manual processes become less scalable. It becomes challenging to manage a large number of expenses or income sources manually.

❖**Dependency on User Engagement**: Manual processes require active user engagement for tasks such as email verification. Users may forget to check their email for verification links, leading to delays in account activation.

❖**Difficulty in Tracking Changes**: Tracking changes made to expenses or income manually can be challenging. Without proper audit trails or version control mechanisms, it's difficult to trace back modifications or identify discrepancies.

❖**Security Risks**: Manual processes for account management and authentication may pose security risks. Weak passwords, sharing credentials, or storing them insecurely can lead to unauthorized access to user accounts.

❖**Lack of Automation Benefits**: Manual processes lack the benefits of automation, such as efficiency, accuracy, and consistency. Without automation, users miss out on features like scheduled expense reminders or automatic categorization.

❖**Difficulty in Reporting and Analysis**: data entry makes it challenging to generate accurate reports or perform in-depth financial analysis. Users may struggle to extract meaningful insights from their financial data due to inconsistencies or errors.

❖**User Frustration**: Cumbersome manual processes may frustrate users, leading to decreased user satisfaction and adoption of the application.

Proposed System

4.1 Scope

❖ **Expense Management**: can add, remove, update, and categorize expenses. Each expense entry may include details such as amount, category, description, and date.

❖ **Income Management**: Users can add, remove, update, and categorize income sources. Income entries may include details such as amount, source, description, and date.

❖ **User Authentication**: Users can register for an account by providing necessary details such as username, email, and password. Registered users can securely log in to access the application's features.

❖ **User Interface**: The application provides an intuitive and user-friendly interface for easy navigation and interaction. Users can efficiently navigate between different sections of the application to manage expenses, income, and other features.

❖ **Analytics**: Expenser offers analytical tools to visualize expenses and income over time. Users can view charts and graphs representing their financial data for better understanding and decision-making.

❖ **Email Verification**: Upon registration, Expenser sends a verification link to the user's email address. Users need to click on the verification link to confirm their email and activate their account.

4.2 Objectives

❖ **Streamlined Financial Management**: Expenser aims to streamline the process of tracking expenses and income sources, providing users with a centralized platform to manage their finances efficiently.

❖**User-Friendly Interface**: The application prioritizes user experience by offering an intuitive and easy-to-navigate interface, ensuring that users can quickly and effectively access the features they need.

❖ **Accurate Data Representation**: Expenser aims to accurately represent users' financial data through charts, graphs, and other visualizations, enabling users to gain meaningful insights into their spending and earning patterns.

❖ **Enhanced Security**: The application prioritizes user security by implementing robust authentication mechanisms, including email verification, to protect user accounts from unauthorized access.

❖ **Customization and Categorization**: Expenser allows users to customize and categorize their expenses and income sources according to their preferences, enabling them to organize their financial data in a way that makes sense to them.

❖ **Data Analysis and Insights**: Expenser provides users with analytical tools to analyse their financial data, identify trends, and make informed decisions about their spending and saving habits.

❖ **Scalability and Reliability**: The application is designed to be scalable and reliable, capable of handling large volumes of data and accommodating a growing user base.

System Planning

5.1Feasibility Study

❖ **Technical Feasibility**: Assess the technical requirements for developing Expenser, including hardware, software, and other resources. Evaluate the suitability of chosen technologies (such as Django for web development) for implementing the required features. Determine if there are adequate technical skills and resources available to develop and maintain Expenser effectively.

❖ **Economic Feasibility**: Evaluate the costs associated with developing, deploying, and maintaining Expenser against the expected benefits. the potential returns from Expenser, considering factors such as user acquisition, revenue generation (if applicable), and cost savings for users. Determine the financial resources required to develop Expenser and assess the availability of funding or investment opportunities.

❖ **Operational Feasibility**: Assess the willingness of users to adopt Expenser and their satisfaction with its features and usability. Evaluate the compatibility of Expenser with existing systems or processes used by potential users. Determine the need for user training and ongoing support to ensure smooth operation and user satisfaction.

❖ **Legal and Regulatory Feasibility**: Identify legal and regulatory requirements related to financial management applications, such as data protection laws (e.g., GDPR) and financial regulations. Assess any potential legal issues related to intellectual property rights, including ownership of code and user data. Ensure compliance with regulations and standards related to user privacy and data security, such as encryption protocols and data retention policies.

❖ **Schedule Feasibility**: Develop a realistic timeline for the development, testing, and deployment of Expenser, considering factors such as resource availability and technical complexity. Define key

milestones and deliverables to track progress and ensure timely completion of the project.

❖ **Risk Analysis**: Identify potential risks and uncertainties that could impact the success of Expenser, such as technical challenges, market competition, and regulatory changes. Develop strategies to mitigate identified risks, such as contingency plans, risk transfer (e.g., insurance), and proactive risk management measures.

❖ **Conclusion**: Based on the findings of the feasibility study, determine whether Expenser is viable and recommend any necessary adjustments or actions to improve its chances of success.

5.2 Requirement Gathering

❖ **Identified Stakeholders**: The first step in gathering information for Expenser was identifying the stakeholders. In this case, my classmates were the primary stakeholders as they would be the users of Expenser for managing their expenses and income sources during their college tenure.

❖ **Conducted Informal Surveys**: To understand the needs and expectations of my classmates regarding a financial management application, I conducted informal surveys and discussions. These interactions provided valuable insights into their financial habits, challenges they faced in managing their expenses, and features they desired in such an application.

❖ **Organized Brainstorming Sessions**: Building on the insights gathered from the informal surveys, I organized group brainstorming sessions with my classmates. These sessions were collaborative and encouraged active participation from everyone involved. We discussed various ideas and feature requests for Expenser, covering aspects such as expense tracking, income management, user interface preferences, and additional functionalities.

❖ **Documented Requirements**: Following the brainstorming sessions, I documented the gathered requirements in a structured manner. The requirements documentation outlined the essential features and functionalities that Expenser would need to fulfill, based on the inputs received from my classmates. This included features like expense tracking, income management, categorization of expenses and income sources, user authentication, and a user-friendly interface.

❖ **Shared and Validated Requirements**: To ensure that the documented requirements accurately reflected the needs and expectations of my classmates, I shared the requirements

documentation and mockups of Expenser's interface with them. This allowed them to provide feedback and suggestions for refinement. Through this iterative process of validation, we were able to fine-tune the requirements and ensure that Expenser would meet the needs of its intended users effectively.

❖ **Finalized Requirements**: After incorporating the feedback received from my classmates and making necessary adjustments, the requirements documentation was finalized. I sought approval from our project supervisor, ensuring that the documented requirements aligned with the scope and objectives of the college project. This marked the completion of the information gathering phase, laying a solid foundation for the subsequent development and implementation stages of Expenser.

Module Specification

Expenser Includes

❖ User-Side Modules:

1. **User Authentication Module**: Allow users to create an account by providing necessary details such as username, email, and password. Enable registered users to securely log in to their accounts. Send a verification link to the user's email for account confirmation.
2. **Expense Management Module**: Allow users to add new expenses with details like amount, category, description, and date. Enable users to delete existing expenses from their account. Allow users to modify details of previously added expenses. Provide users with a search functionality to find specific expenses based on various criteria.
3. **Income Management Module**: Allow users to add new sources of income with details like amount, source, description, and date. Enable users to delete existing income sources from their account. Allow users to modify details of previously added income sources.

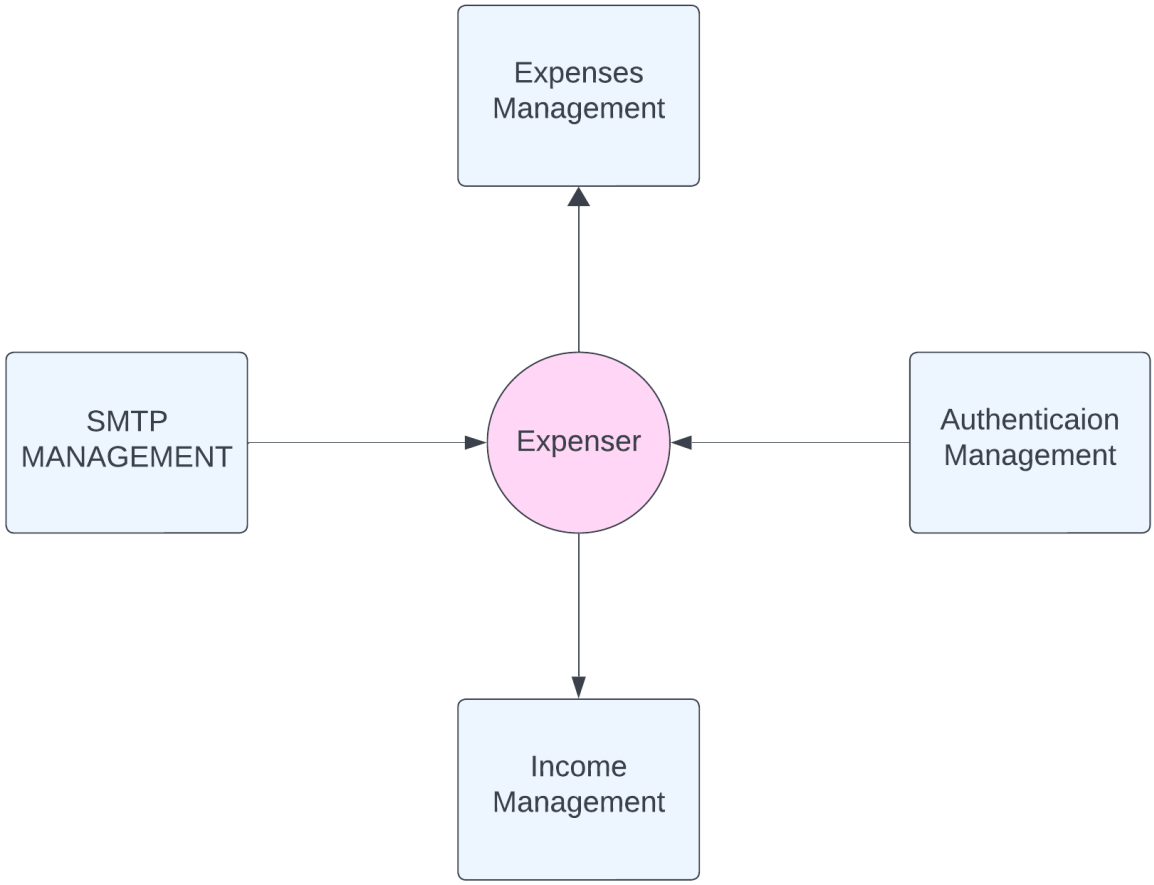
❖ User-Side Modules:

1. **Category Management Module**: Allow admin users to add new categories for expenses. Enable admin users to delete existing expense categories. Allow admin users to modify details of previously added expense categories.
2. **Income Source Management Module**: Income Source: Allow admin users to add new sources of income. Enable admin users to delete existing income sources. Allow admin users to modify details of previously added income sources.

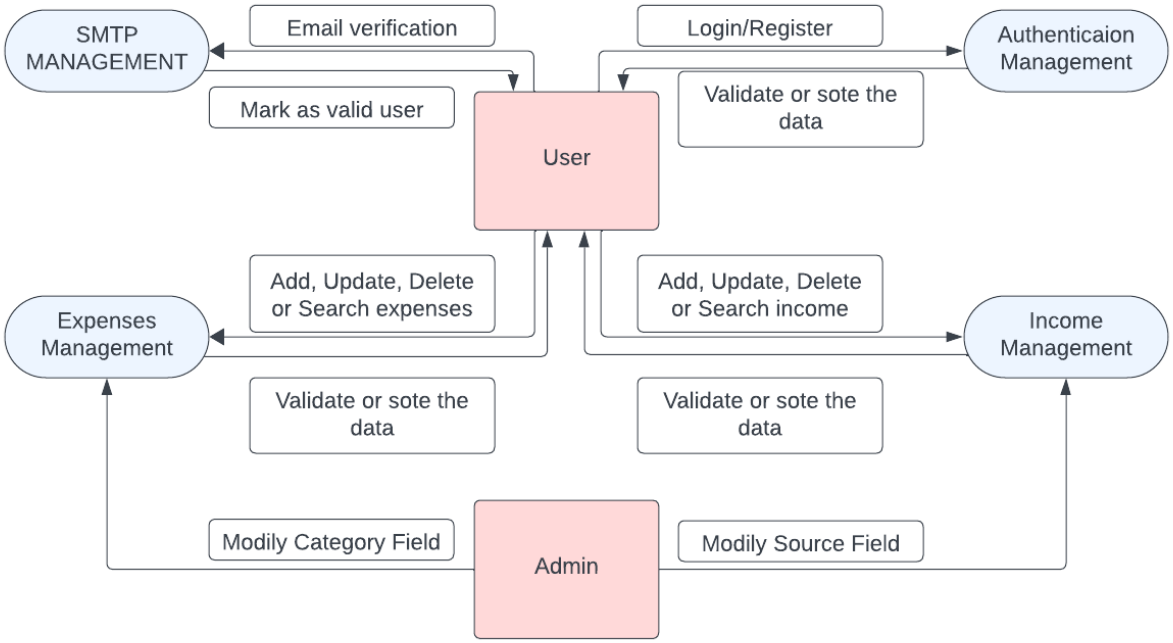
Detail Planning

7.1 DFD for Expenser

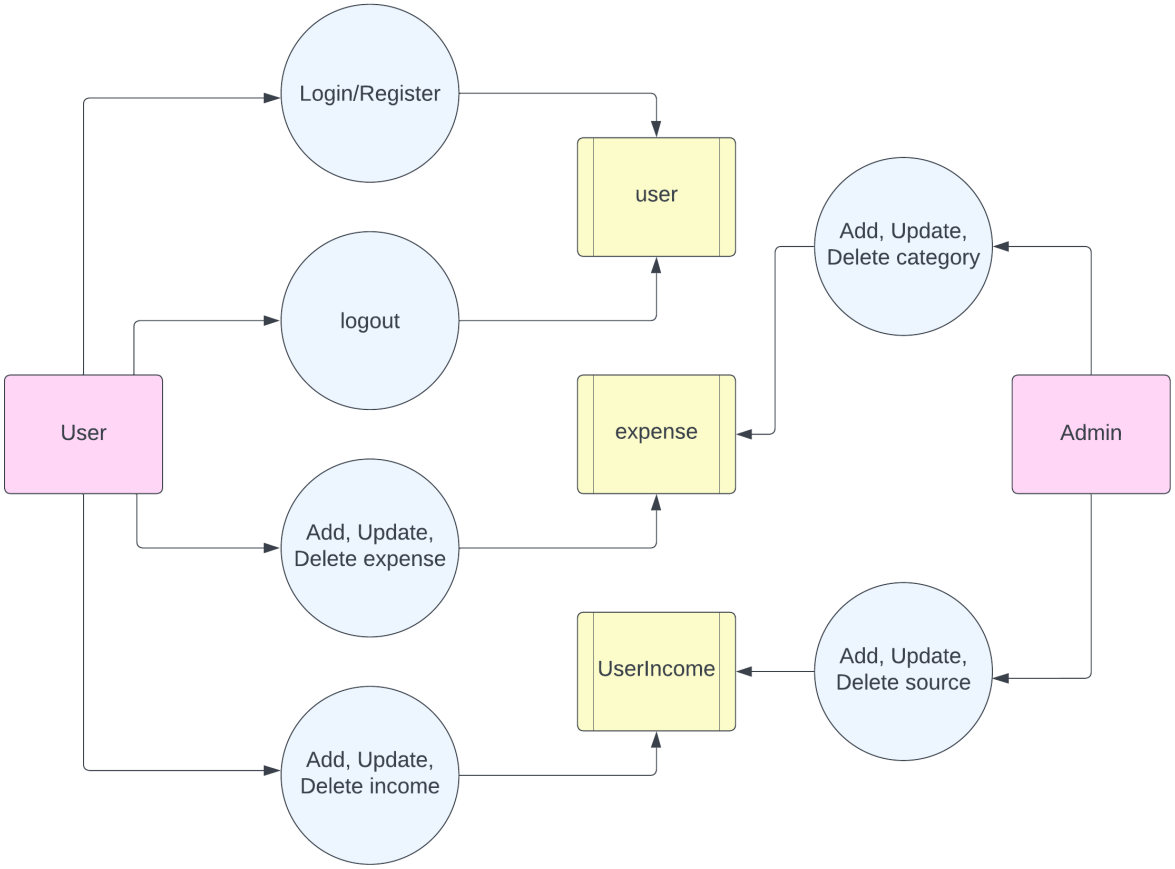
❖ Level 0:



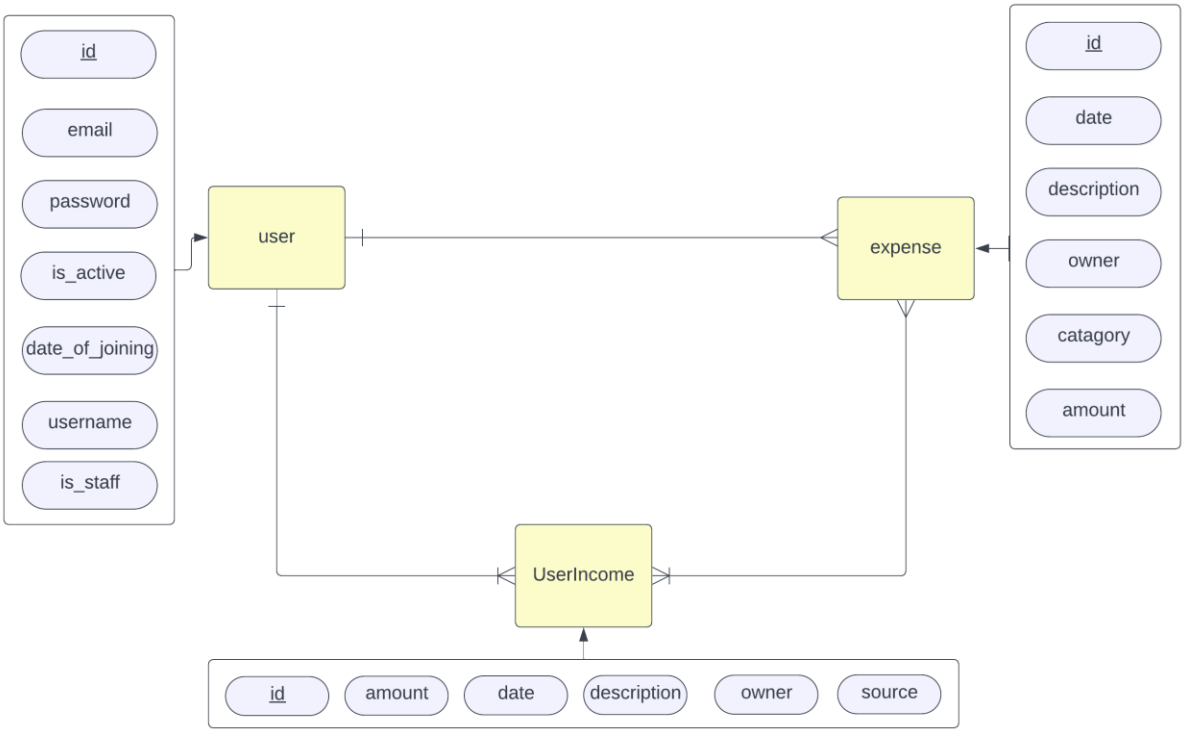
❖ Level 1:


SMTP MANAGEMENT
Email verification
Mark as valid user
Login/Register
Validate or sote the data
Authenticaion Management
User
Add, Update, Delete or Search expenses
Validate or sote the data
Add, Update, Delete or Search income
Validate or sote the data
Expenses Management
Income Management
Modily Category Field
Modily Source Field
Admin

❖ Level 2:


Login/Register
user
Add, Update,
Delete category
logout
User
expense
Admin
Add, Update,
Delete expense
UserIncome
Add, Update,
Delete source
Add, Update,
Delete income

7.2 Entity Relationship Diagram



7.3 Database Design

❖ user:

Field Name	Data Type	Constrain
id	integer	Primary key
email	varchar(254)	
password	varchar(128)	
is_active	boolean	
date_of_joining	date	
username	varchar(150)	
is_staff	boolean	

❖ expense:

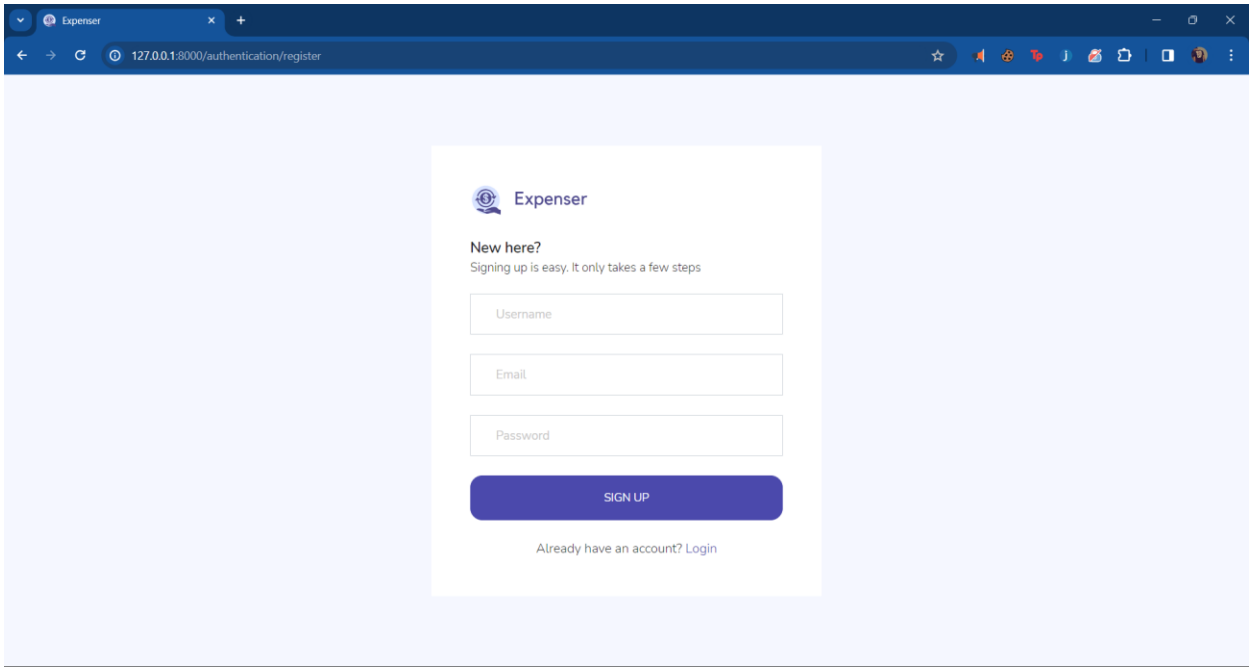
Field Name	Data Type	Constrain
id	integer	Primary key
date	date	
description	text	
owner	integer	
category	varchar(266)	
amount	double precision	

❖ UserIncome:

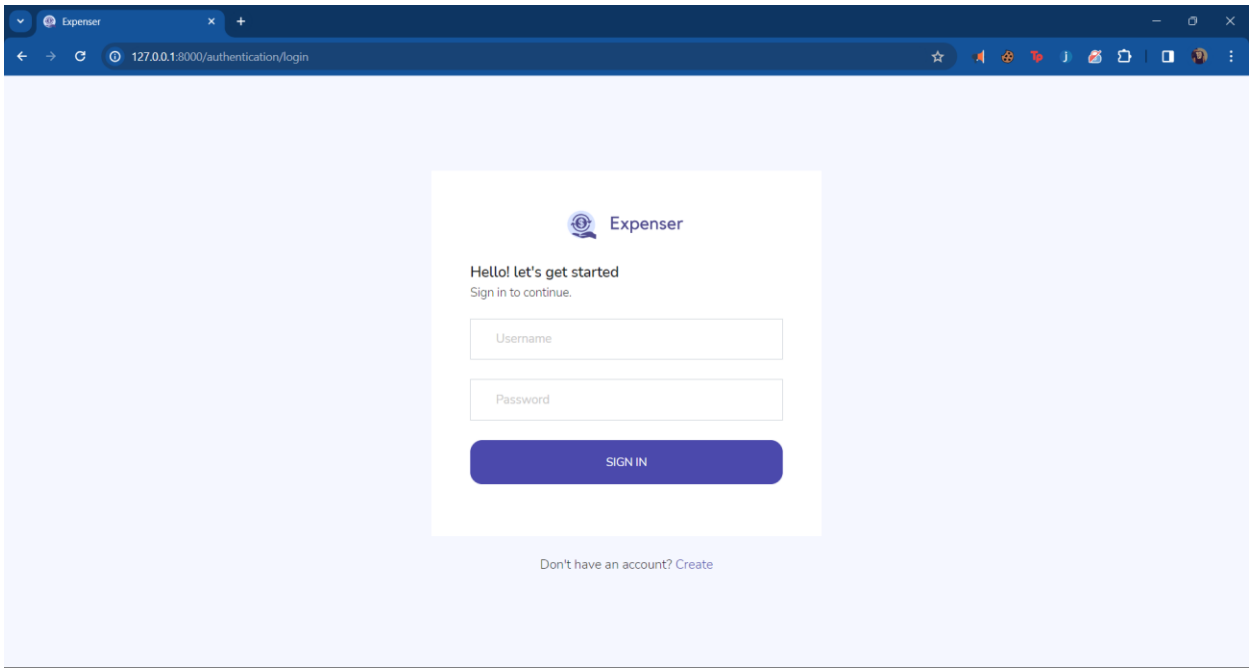
Field Name	Data Type	Constrain
id	integer	Primary key
date	date	
description	text	
owner	integer	
source	varchar(266)	
amount	double precision	

System Design

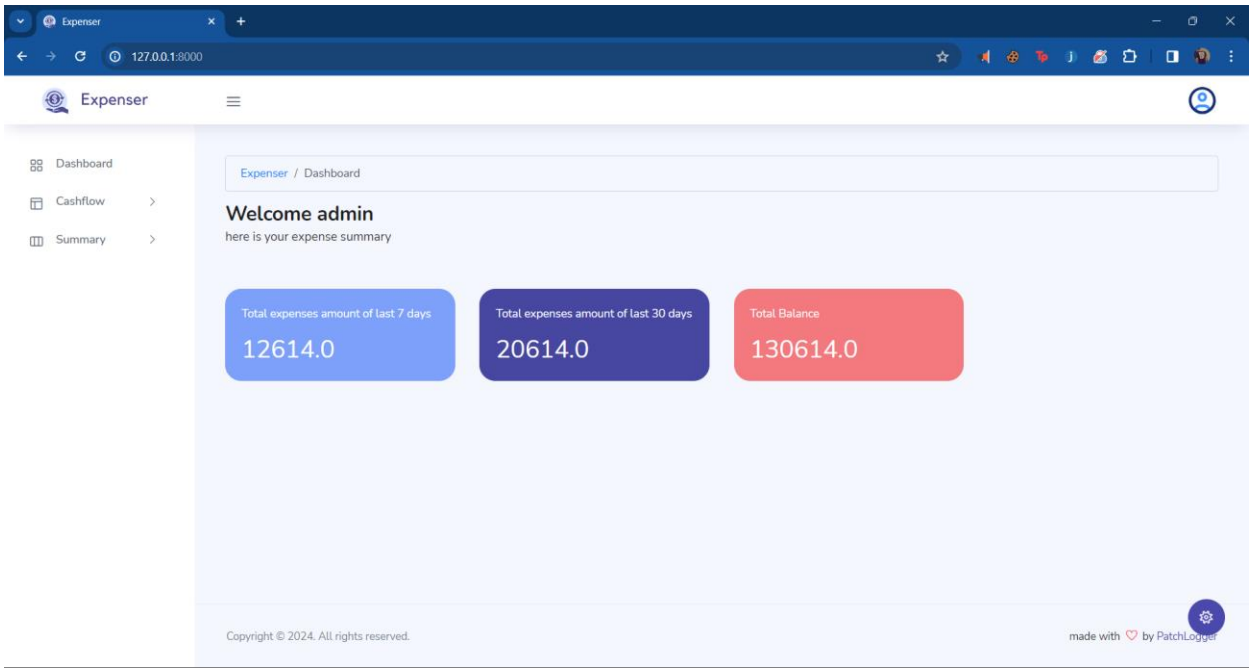
❖ Registration Page:



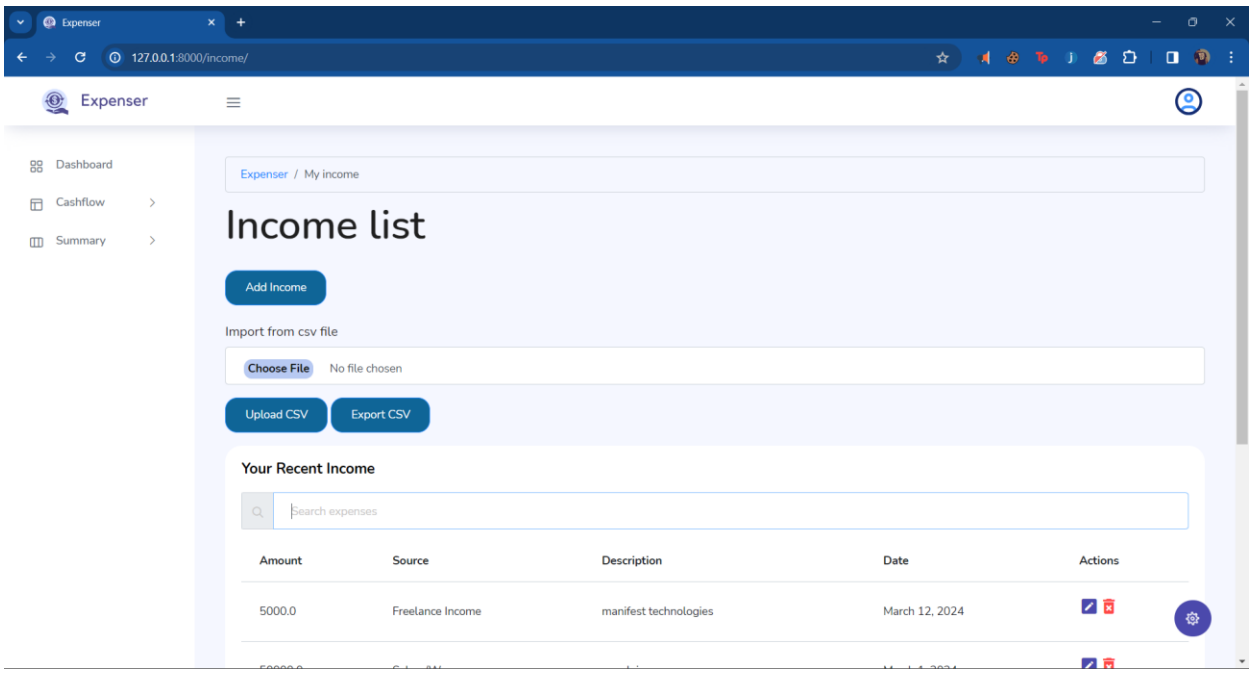
❖ Login Page:



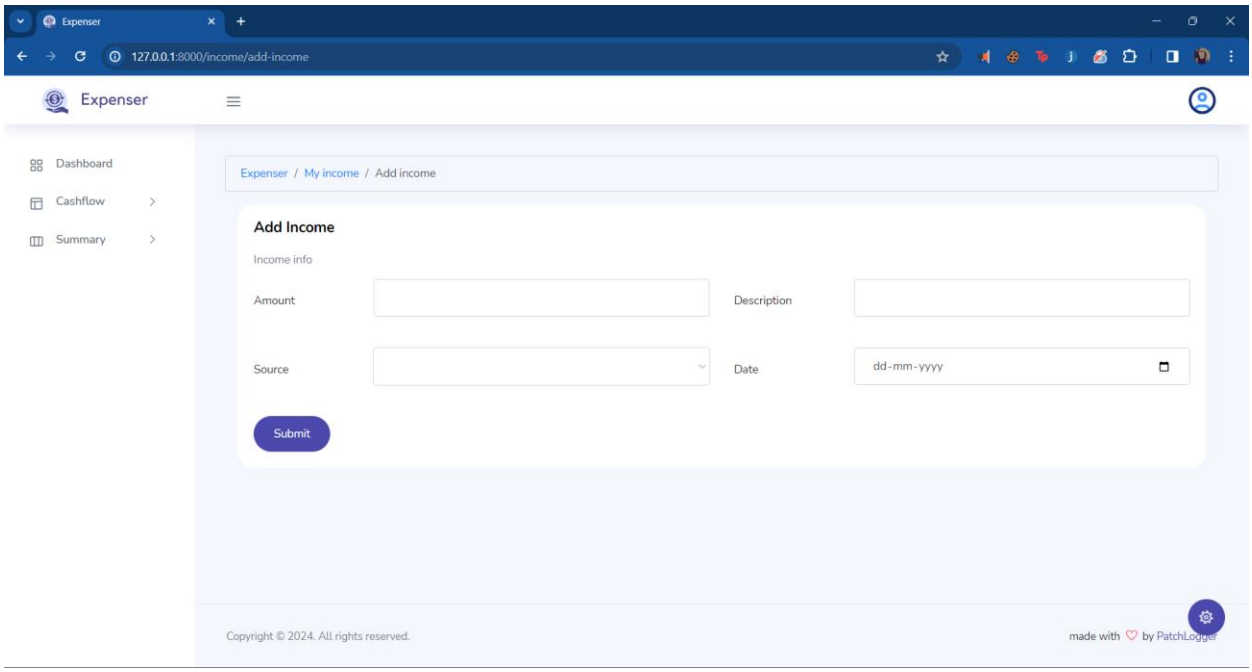
❖ Dashboard:



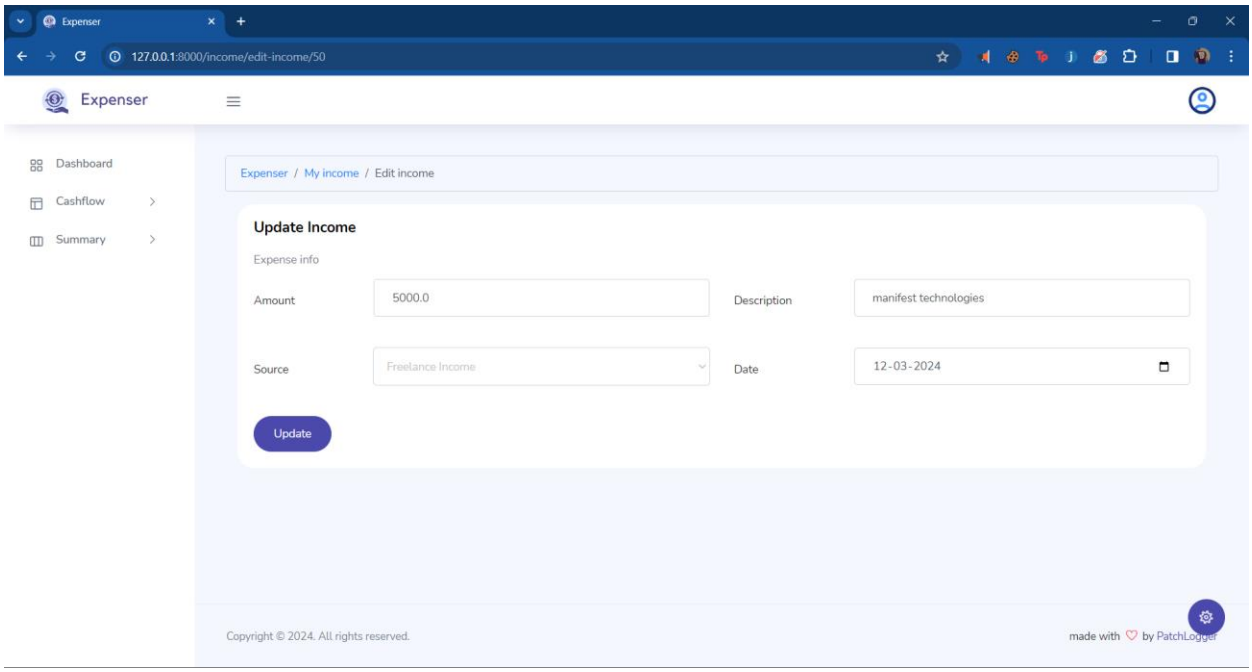
❖ Income List:



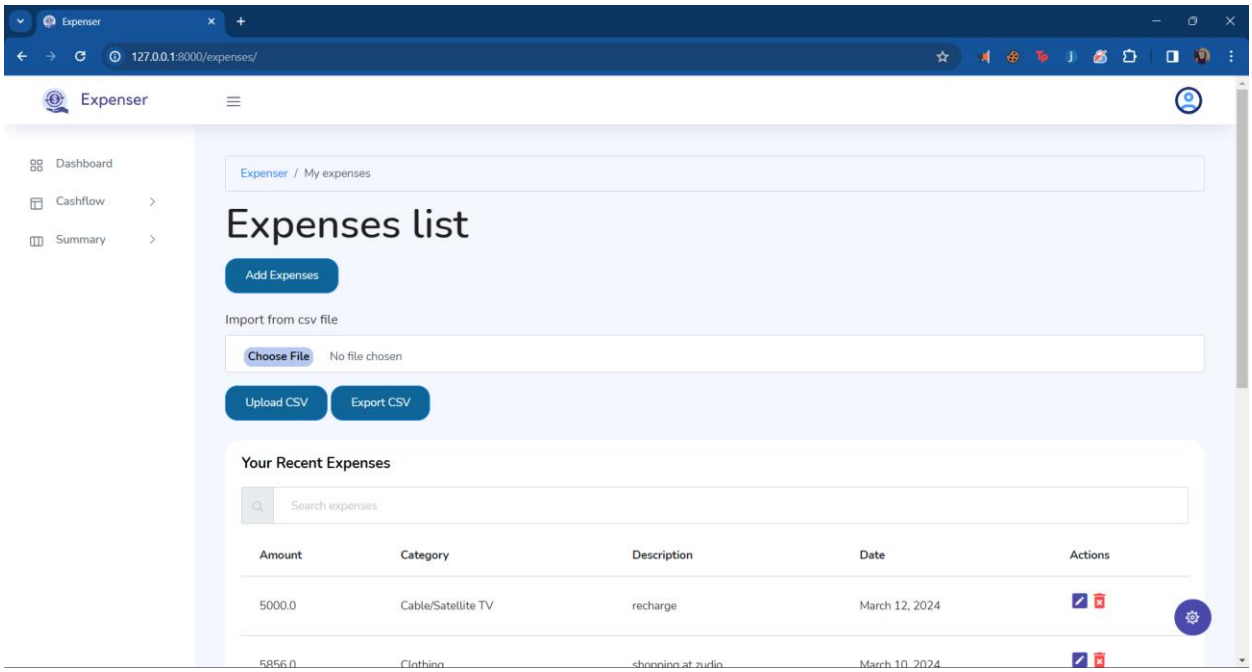
❖Add Income:



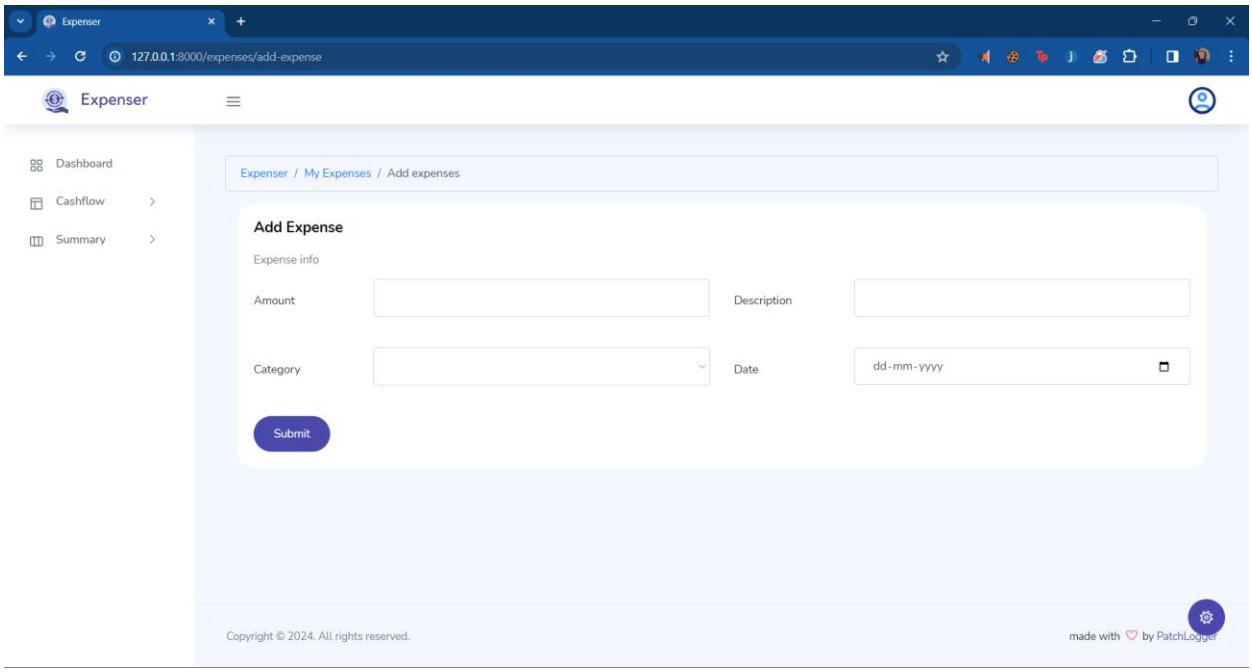
❖ Update Income:



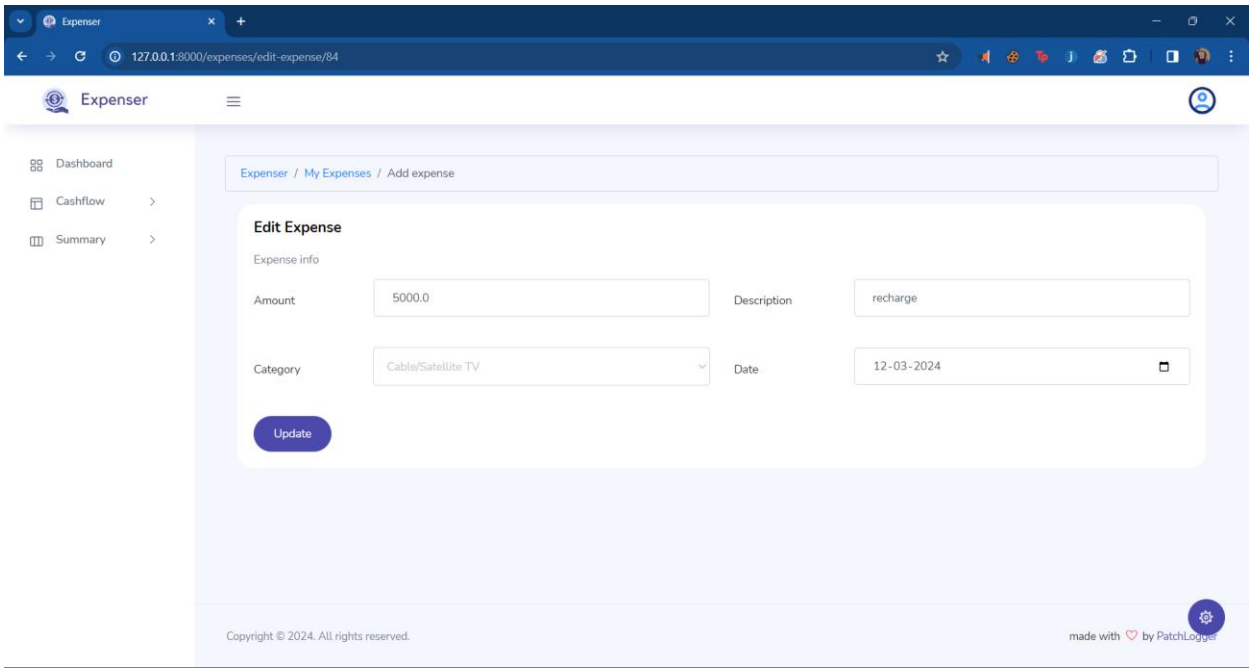
❖ Expense List:



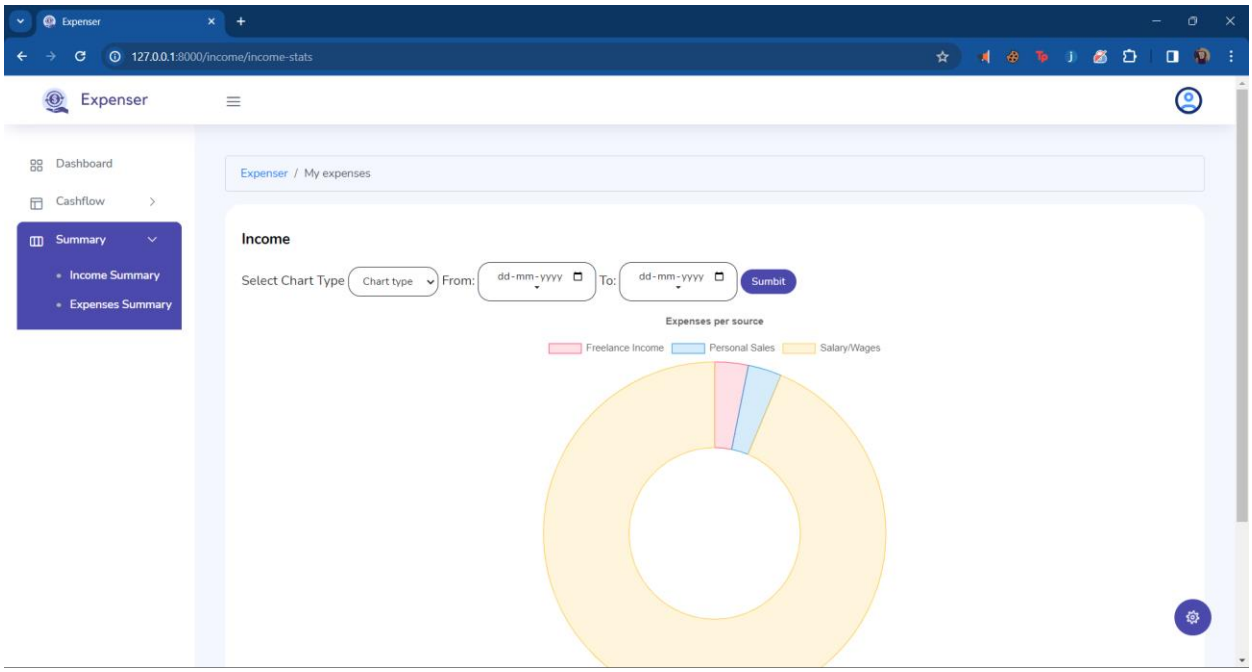
❖ Add Expense:



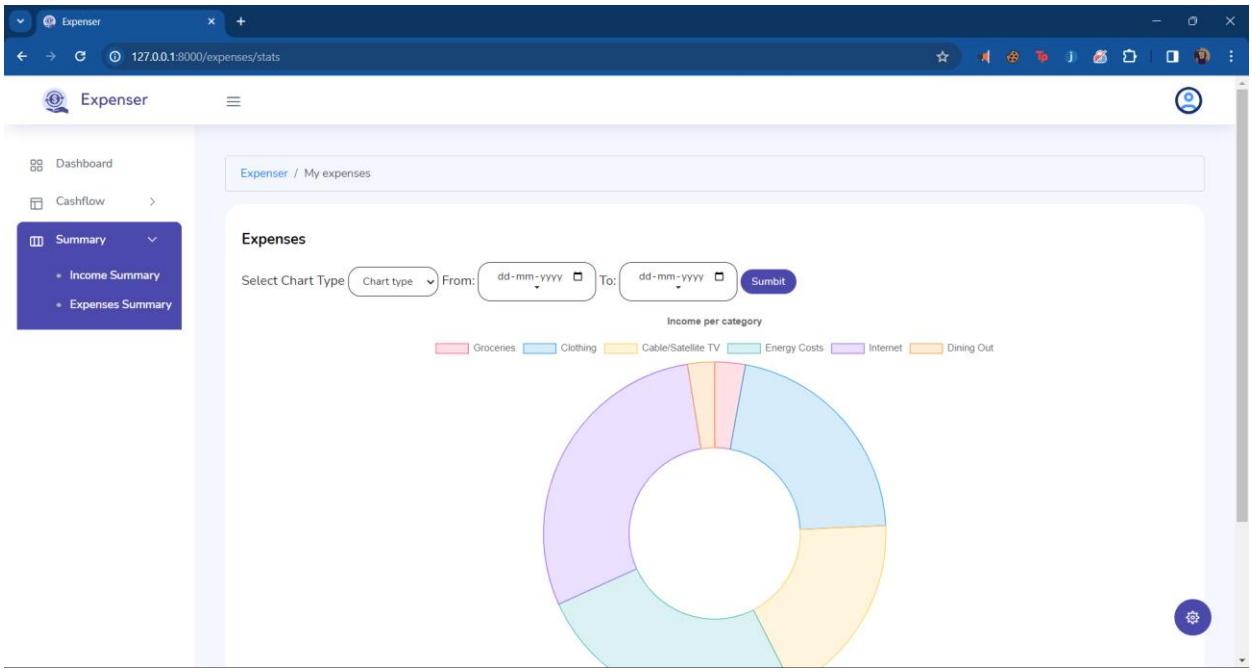
❖ Update Expense:



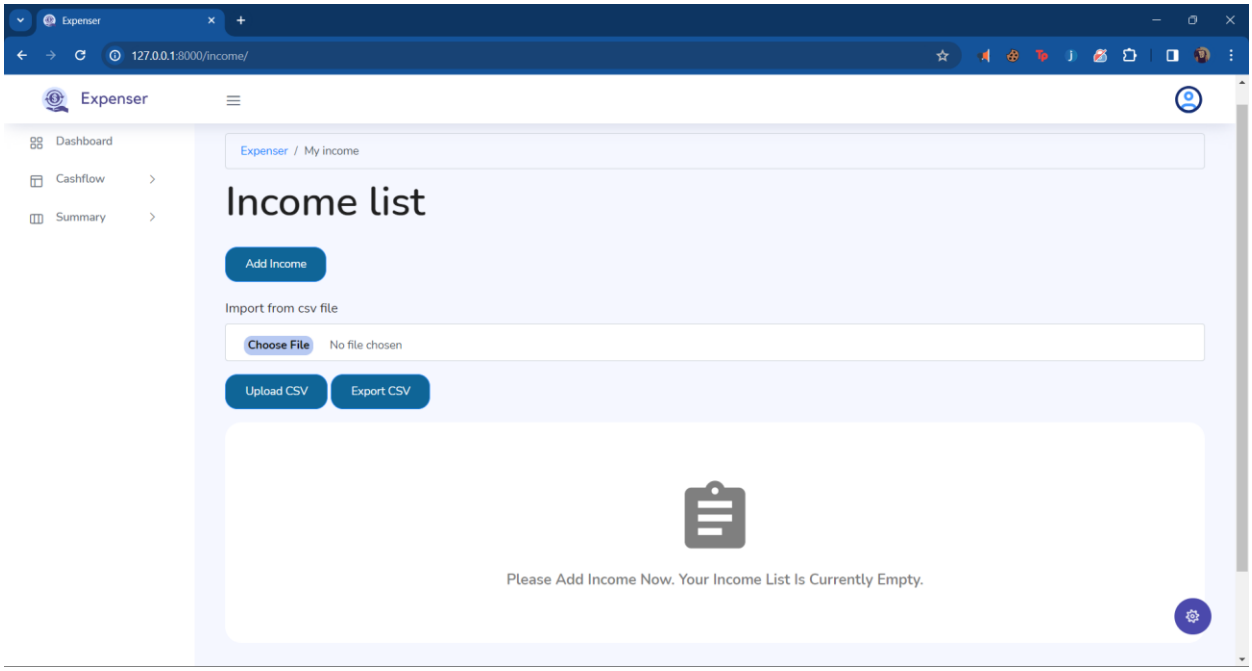
❖ Income Summary:



❖ Expense Summary:



❖ Empty List:



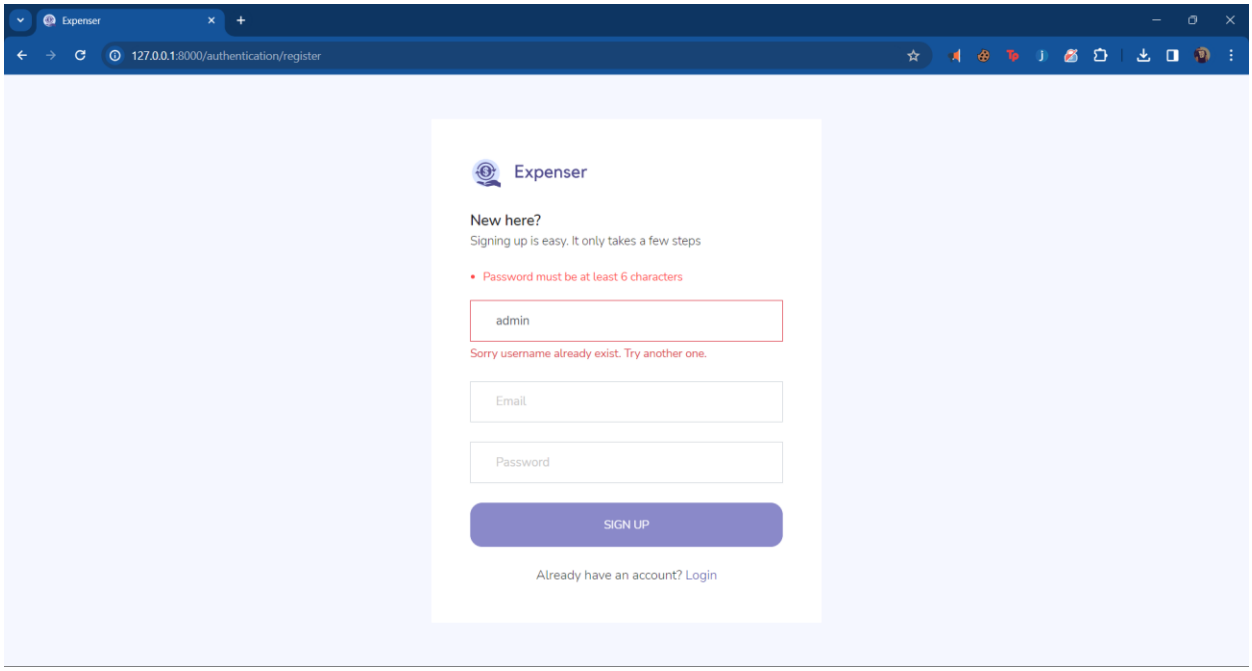
System Testing

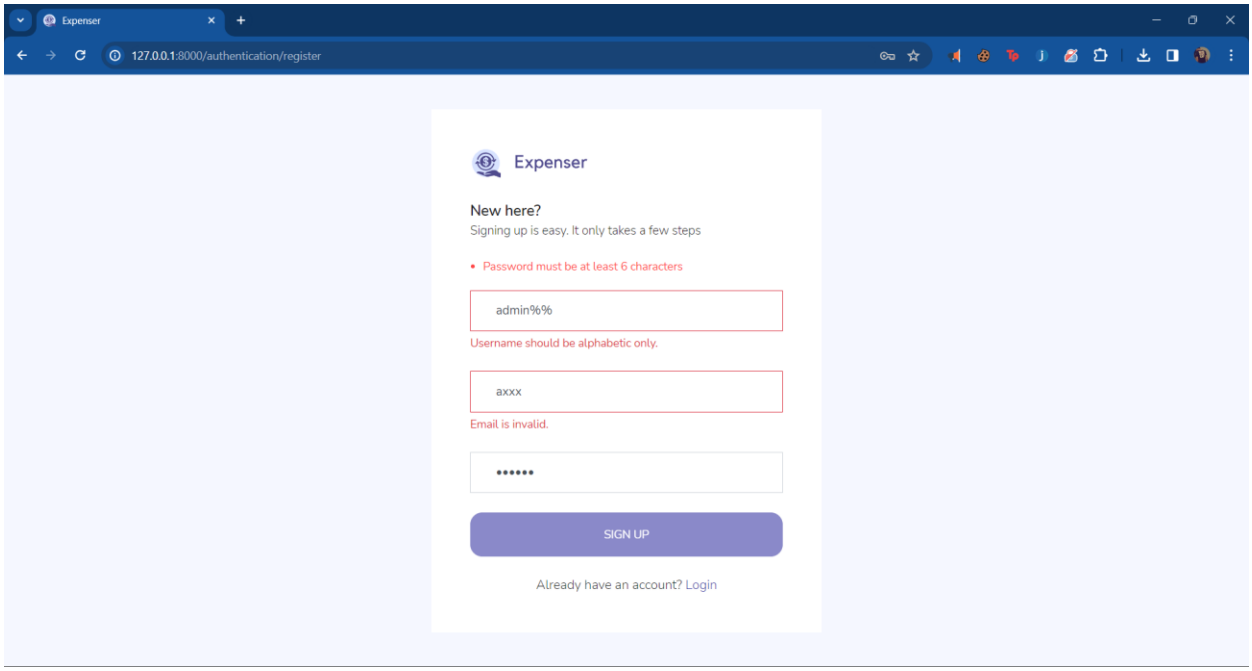
9.1 Unit Testing

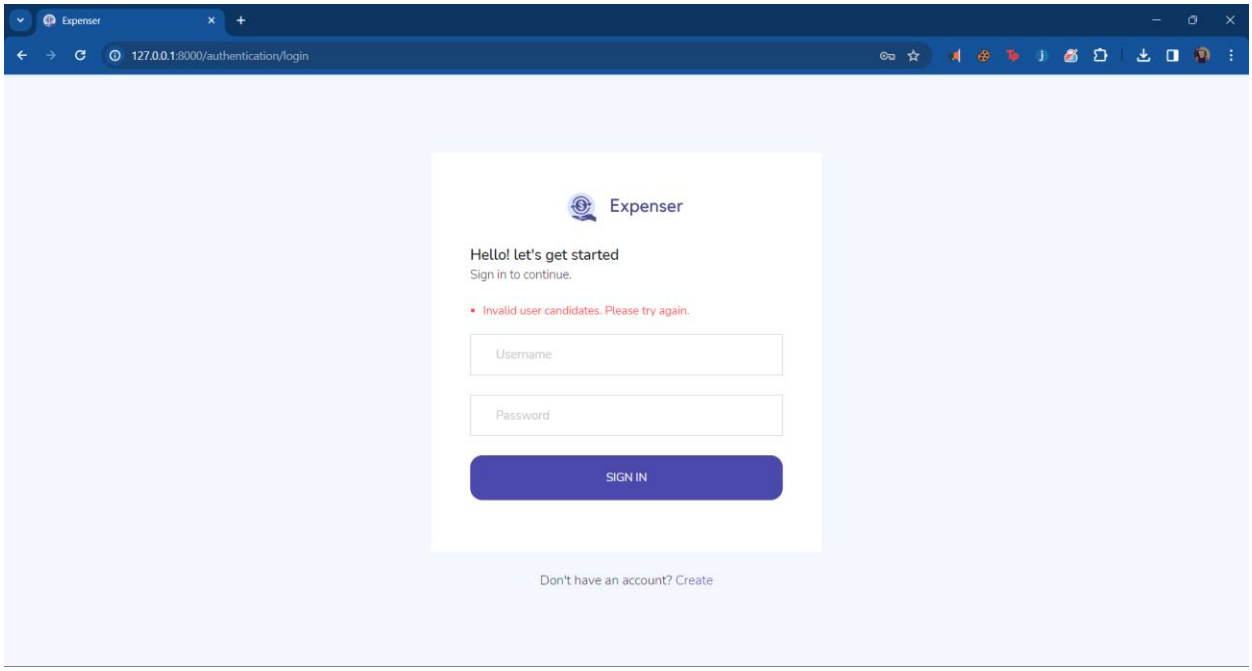
This is the lowest level of testing that is conducted to remove syntax & logic errors from a single unit. Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. This phase involves testing collection of modules, which have been integrated into sub-system. These testes for problems that arise form component interactions. This testing should begin as soon as usable version of some of the system components is available.

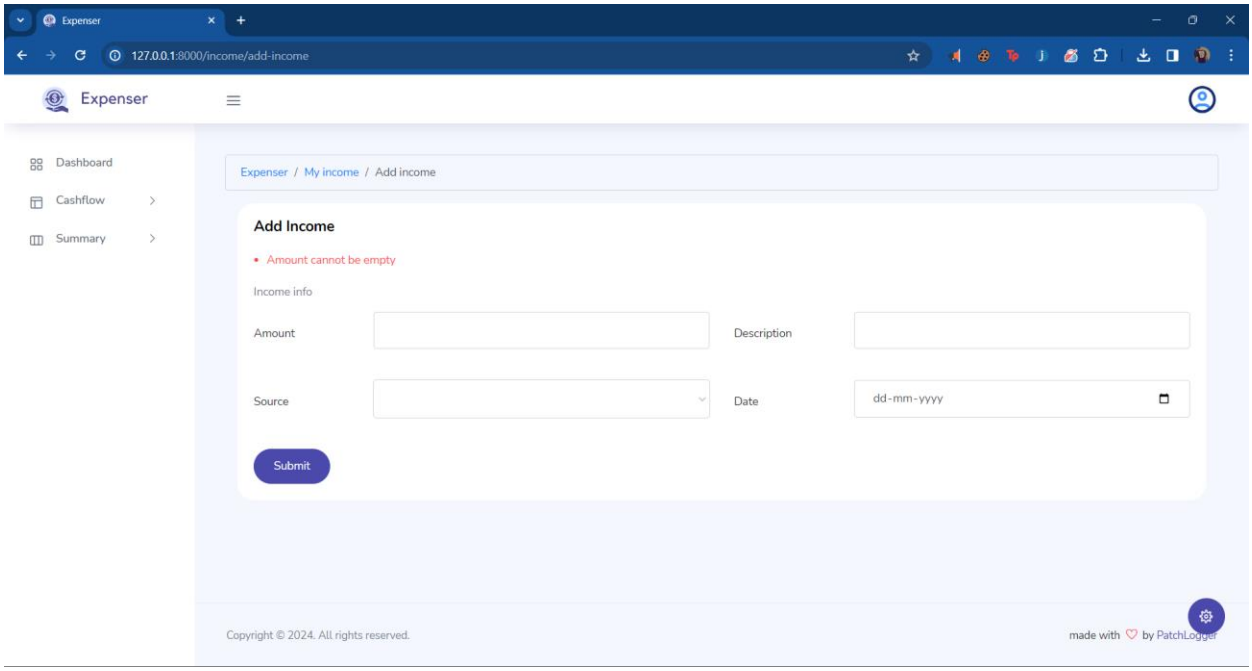
9.2 Integration Testing

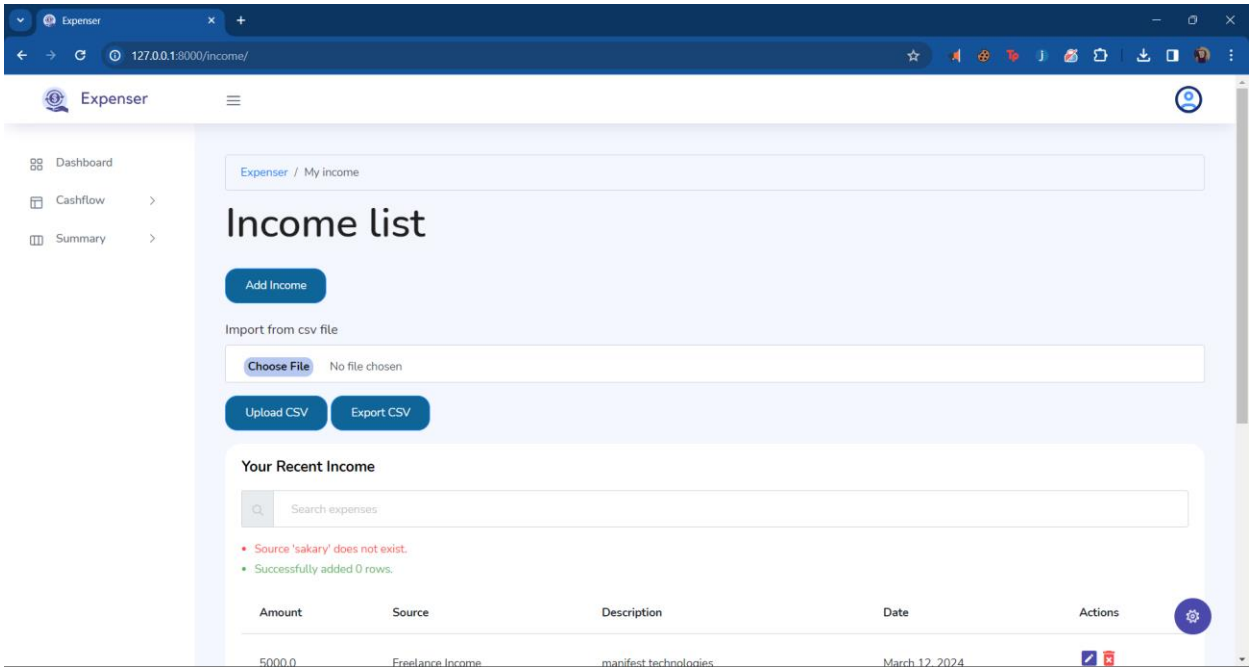
This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system customer rather than simulated test data. Acceptance testing may reveal errors & omissions in the system requirements definition because the real data exercise the system in different ways from the test data. It may also reveal requirements problems where the system's facilities do not really meet the user's needs or the system performance is unacceptable.

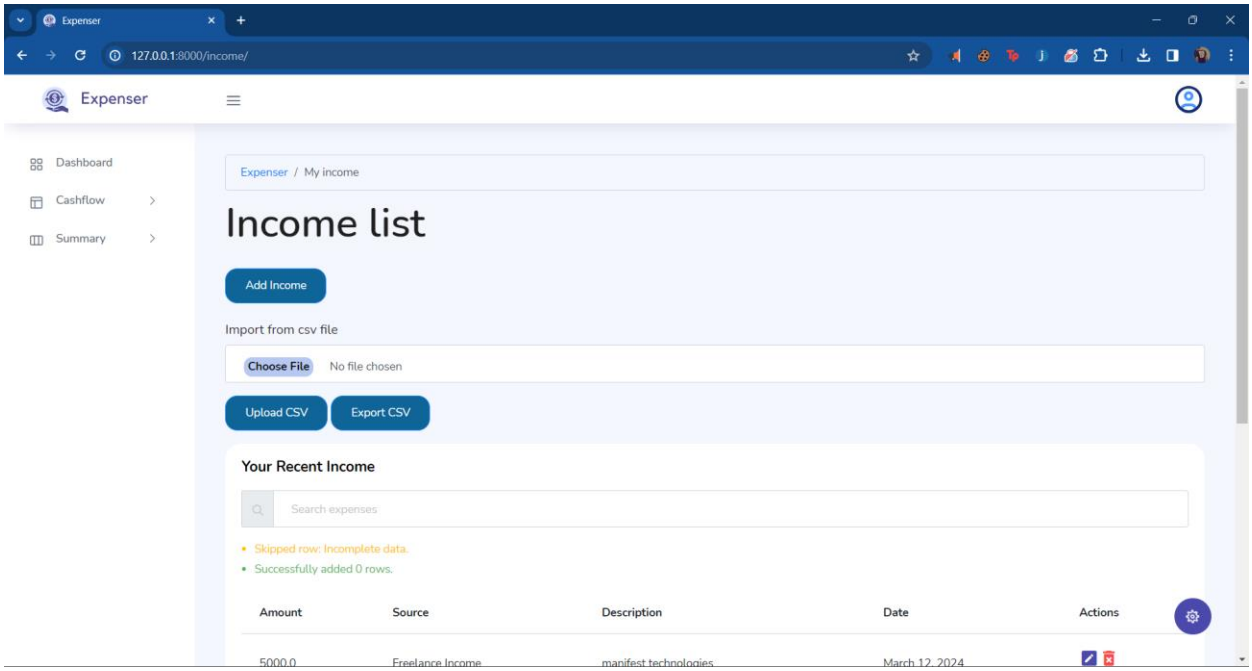

Expenser
127.0.0.1:8000/authentication/register
Expenser
New here?
Signing up is easy. It only takes a few steps
Password must be at least 6 characters
admin
Sorry username already exist. Try another one.
Email
Password
SIGN UP
Already have an account? Login

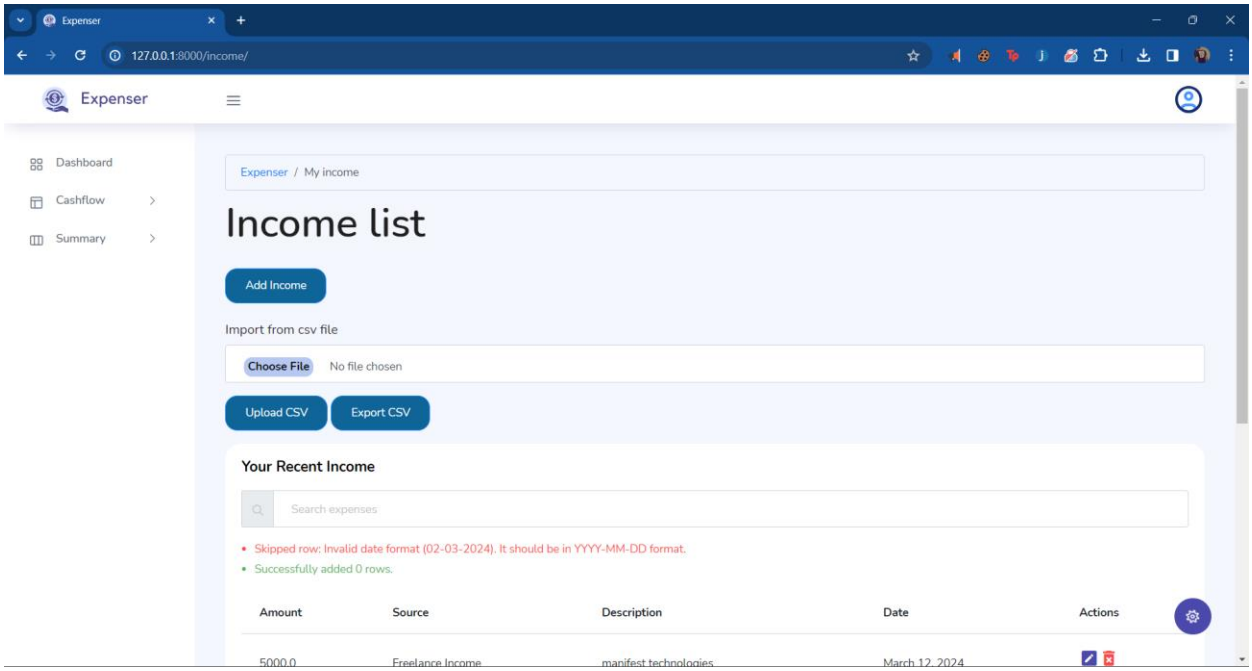

Expenser
127.0.0.1:8000/authentication/register
Expenser
New here?
Signing up is easy. It only takes a few steps
Password must be at least 6 characters
admin%%
Username should be alphabetic only.
axxx
Email is invalid.
SIGN UP
Already have an account? Login

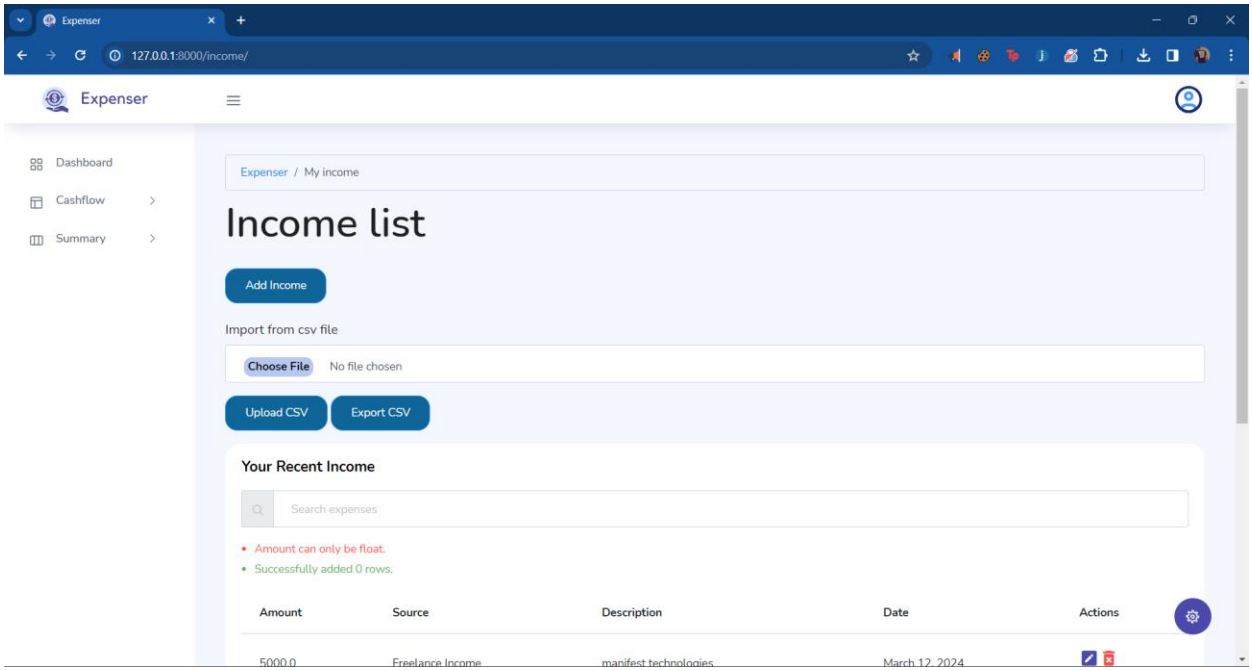

Expenser
127.0.0.1:8000/authentication/login
Expenser
Hello! let's get started
Sign in to continue.
Invalid user candidates. Please try again.
Username
Password
SIGN IN
Don't have an account? Create


Expenser
127.0.0.1:8000/income/add-income
Expenser
Dashboard
Cashflow
Summary
Expenser / My income / Add income
Add Income
Amount cannot be empty
Income info
Amount
Description
Source
Date
dd-mm-yyyy
Submit
Copyright © 2024. All rights reserved.
made with by PatchLogger


Expenser
127.0.0.1:8000/income/
Expenser
Dashboard
Cashflow
Summary
Expenser / My income
Income list
Add Income
Import from csv file
Choose File
No file chosen
Upload CSV
Export CSV
Your Recent Income
Search expenses
Source 'sakary' does not exist.
Successfully added 0 rows.
Amount
Source
Description
Date
Actions
5000.0
Freelance Income
manifest technologies
March 12, 2024


Expenser
127.0.0.1:8000/income/
Expenser
Dashboard
Cashflow
Summary
Expenser / My income
Income list
Add Income
Import from csv file
Choose File
No file chosen
Upload CSV
Export CSV
Your Recent Income
Search expenses
Skipped row: Incomplete data.
Successfully added 0 rows.
Amount
Source
Description
Date
Actions
5000.0
Freelance Income
manifest technologies
March 12, 2024


Expenser
127.0.0.1:8000/income/
Expenser
Dashboard
Cashflow
Summary
Expenser / My income
Income list
Add Income
Import from csv file
Choose File
No file chosen
Upload CSV
Export CSV
Your Recent Income
Search expenses
Skipped row: Invalid date format (02-03-2024). It should be in YYYY-MM-DD format.
Successfully added 0 rows.
Amount
Source
Description
Date
Actions
5000.0
Freelance Income
manifest technologies
March 12, 2024


Expenser
127.0.0.1:8000/income/
Expenser
Dashboard
Cashflow
Summary
Expenser / My income
Income list
Add Income
Import from csv file
Choose File
No file chosen
Upload CSV
Export CSV
Your Recent Income
Search expenses
Amount can only be float.
Successfully added 0 rows.
Amount
Source
Description
Date
Actions
5000.0
Freelance Income
manifest technologies
March 12, 2024

References

❖ Websites:

- https://djangopackages.org/
- https://docs.djangoproject.com/
- https://www.chartjs.org/
- https://getbootstrap.com/docs/
- https://stackoverflow.com/
- https://www.postgresql.org/docs/