**CAN THO UNIVERSITY**

**COLLEGE OF INFORMATION AND COMMUNICATOIN TECHNOLOGY**



**SPECIALIZED DISSERTAION**

**INFORMATION TECHNOLOGY**

**FINANCE MANAGEMENT WEBSITE USING DJANGO FRAMEWORK**

**Student : Luu Thai Hoa**

**Student ID: B2014194**

**Class : K46**

**Advisor : Dr. Thai Minh Tuan**

**Can Tho, 11/2023**

**CAN THO UNIVERSITY**

**COLLEGE OF INFORMATION AND COMMUNICATOIN TECHNOLOGY**



**SPECIALIZED DISSERTAION**

**INFORMATION TECHNOLOGY**

**FINANCE MANAGEMENT WEBSITE USING DJANGO FRAMEWORK**

**Student : Luu Thai Hoa**

**Student ID: B2014194**

**Class : K46**

**Advisor : Dr. Thai Minh Tuan**

**Can Tho, 11/2023**

# **EVALUATION OF ADVISOR**

Advisor

Thai Minh Tuan

# **ACKNOWLEDGEMENTS**

I want to express my gratitude for DR. Thai Minh Tuan’s guidance during the completion of my specialized dissertation. Then I would like to thank the lecturers of College Of Information and Communication Technology who taught me knowledge when I studying.

I am very grateful for my family’s support, also my friends for solving problems together when I was doing research in Can Tho University.

Sincerely,

Can Tho, 11/22/2023

Luu Thai Hoa

**Table of Contents**

[**EVALUATION OF ADVISOR** 3](#_Toc151818914)

[**ACKNOWLEDGEMENTS** 4](#_Toc151818915)

[**LIST OF PICTURE** 6](#_Toc151818916)

[**LIST OF TABLE** 8](#_Toc151818917)

[**LIST OF ABBREVIATIONS** 9](#_Toc151818918)

[**ABSTRACT** 10](#_Toc151818919)

[**PART 1: INTRODUCTION** 1](#_Toc151818920)

[**I.** **Problems** 1](#_Toc151818921)

[**II.** **Problems Solving History** 1](#_Toc151818922)

[**III.** **Objectives and scope** 2](#_Toc151818923)

[**IV.** **Research Content** 2](#_Toc151818924)

[**V.** **Outline** 2](#_Toc151818925)

[**PART 2: RESEARCH CONTENT** 3](#_Toc151818926)

[**CHAPTER 1 : BACKGROUND** 3](#_Toc151818927)

[**1.1.** **Background** 3](#_Toc151818928)

[**CHAPTER 2: DESIGN AND IMPLEMENTATION** 6](#_Toc151818932)

[**2.1.** **Overview of the website** 6](#_Toc151818933)

[**2.2.** **Main Functions** 7](#_Toc151818934)

[**2.3.** **Use Cases Diagrams** 8](#_Toc151818935)

[**2.4.** **Database Design** 10](#_Toc151818939)

[**2.5.** **Detail Of Functions In The System** 11](#_Toc151818942)

[**CHAPTER 3: TESTING AND EVALUATION** 26](#_Toc151818970)

[**3.1.** **Target of testing** 26](#_Toc151818971)

[**3.2.** **Testing senario** 26](#_Toc151818972)

[**3.3.** **Result of testing** 27](#_Toc151818973)

[**PART 3 : CONCLUSION AND FUTURE WORK** 28](#_Toc151818974)

[**I.** **Conclusion** 28](#_Toc151818975)

[**II.** **Future Work** 28](#_Toc151818976)

[**REFERENCES** 29](#_Toc151818977)

# **LIST OF PICTURE**

[Picture 1. how api work 3](#_Toc151817737)

[Picture 2. Django logo 4](#_Toc151817738)

[Picture 3. how django work 5](#_Toc151817739)

[Picture 4. Use case diagram of guest 8](#_Toc151817744)

[Picture 5. Use case diagram of user 8](#_Toc151817745)

[Picture 6. Use case diagram of admin 9](#_Toc151817746)

[Picture 7. Database design 10](#_Toc151817748)

[Picture 8. Create expense table using models 10](#_Toc151817749)

[Picture 9. Flowchart of sign up function 11](#_Toc151817751)

[Picture 10. Flowchart of sign in function 12](#_Toc151817752)

[Picture 11. api flowchart 13](#_Toc151817753)

[Picture 12. login interface 14](#_Toc151817754)

[Picture 13. sign up interface 14](#_Toc151817755)

[Picture 14. error message with taken username 15](#_Toc151817756)

[Picture 15. error message with take email 15](#_Toc151817757)

[Picture 16. account created when all condition meet 16](#_Toc151817758)

[Picture 17. side bar of the website 16](#_Toc151817759)

[Picture 18. Expense interface 17](#_Toc151817760)

[Picture 19. adding an expense 17](#_Toc151817761)

[Picture 20. edit or delete an expense 18](#_Toc151817762)

[Picture 21. export to pdf file 18](#_Toc151817763)

[Picture 22. export to excel file 19](#_Toc151817764)

[Picture 23. export to csv file 19](#_Toc151817765)

[Picture 24. Income interface 19](#_Toc151817766)

[Picture 25. add income 20](#_Toc151817767)

[Picture 26. edit or delete income 20](#_Toc151817768)

[Picture 27. export to PDF file 21](#_Toc151817769)

[Picture 28. export to excel file 21](#_Toc151817770)

[Picture 29. export to csv file 22](#_Toc151817771)

[Picture 30. chart of expenses 22](#_Toc151817772)

[Picture 31.chart of incomes 23](#_Toc151817773)

[Picture 32. stock price view 23](#_Toc151817774)

[Picture 33. stock simulator interface 24](#_Toc151817775)

[Picture 34. buying stock interface 24](#_Toc151817776)

[Picture 35. edit or delete bought stock 25](#_Toc151817777)

**LIST OF TABLE**

Table 1. Main functions of the system……………………………………………………...7

Table 2. list of installed models……………………………………………………….11

Table 3. Result of testing……………………………………………………………...27

# **LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviations** | **Description** |
| API | Application Programming Interface |
| Usecase | Use Case Diagram |
| UI/UX | User interface, user experience |

# **ABSTRACT**

Nowadays, finance is one of the most important thing a person had to manage, from college fee to house fee, meal, gas,…. For a student, if they manage their finance incorrectly, they could run out of money before the month end. From that, with the aim to help student manage their finance better, this dissertation develop Income management website which is a website where user can enter their incomes and expenses for better management, it also provide stock investment simulator for student to practise investing.

The website is implemented with main function such as manage expenses, income, invest simulate, stock chart. The front-end of the website is using HTML, CSS. Back-end is using Django and JavaScript. Hopefully this website can help user in manage their finance better

# **PART 1: INTRODUCTION**

1. **Problems**

Finance has always been a part of our life ever since money was invented. From meal fees to goods, house fees,… And we are living in the Fourth Industrial Revolution, managing our incomes and expenses has become more important than ever before. With the benefit of bringing a lot of convenience and speeding up our lifestyle, Industry 4.0 also makes our life more complicated. There are much more choices for products, services, how to spend and it requires human to have skills to manage their finance to not meet any problems in the future. This lead to a lot of people cant catch up with the speed of life, and therefore make mistake, causing ton of problems in the present and in the future. And another problem is that most people don’t know how to invest in stock, data from 2023 show that 92% of vietnamese dont have a stock trading account. Stock is the best way to have life insurance, for example : if a person start investing at 25, each month he invest 20 dollar and increase it 10% each year, with the avage market grow of 10%, he would make 650,000 dollar at the age of 60, and the money he spend is only 76,000 dollar for investing, that is 850% revenue.

“Finance Management Website” is created with the desire of helping individual, especially student in managing their finance more easily, create condition for them to focus more on studying and also introduce a stock trading simulator for student to pratice long-term investment.

1. **Problems Solving History**

There are a lot of apps that design for user to manage their finance

* Money Lover: This have basic of mange finance, and one of the features that this application offers is the savings account, with the ability to directly link to other bannks, making it convenient for seamless transactions
* PocketGuar: With this application, you can show the picture of bills and it can help you calculate and analysis through chart
* …

But most of it not have stock trading simulator, and some of them need you to pay to work.

1. **Objectives and scope**

The website provides main function for users to manage their expenses, incomes, also provide export function for user to export their expenses or incomes. It also can show stock in real-time and can have a simulator that can help you practice buy stock in the past. The scope of the study is: researching the problem of financing, why people don’t trading stock, solving the problem with implementing website with friendly UI/UX

1. **Research Content**

**1. Methods**

- Analyzing requirements: research problems on internet.

- Implementing: Django framwork, HTML, CSS, Javascript, Postgresql, API.

**2. Solutions:**

- Using analyzed data to designing the website, building website using HTML, CSS, Javascript, Django, Postgresql, API.

- Softwares: Google Chrome for web browser, VScode; Framework: Django

1. **Outline**

**Part 1 – Introduction:**

This part introduces research problems, the purpose of study, the scope of the study and research content. At the end of this chapter, there is a conten introduction of each chapter.

**Part 2 – Research Content:**

Including three main parts:

**Chapter 1:** Background

General information about the study and main functions of the system.

**Chapter 2:** Design and implementation

Introduction of UI/UX designs, models and implementation, describing the eachnologies used in the website, use casa diagrams, database and details of function in the website.

**Chapter 3:** Presentation the testing goals, scenarios and the testing results.

**Part 3 – Conclusion and future work:**

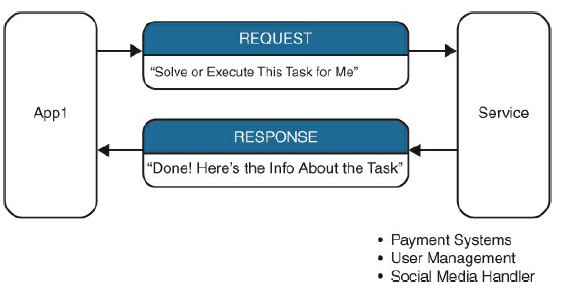
Summary the result and recommends improvements in the future.

**PART 2: RESEARCH CONTENT**

**CHAPTER 1 : BACKGROUND**

* 1. **Background**
     1. **­­API**

API ( Application programming interface ) is a set of interaction rules designed for applications or components of an application to communicate with each other. The purpose of an API is to simplify the process of interaction to communicate with each other. The purpose of an API is to simplify the process of interaction and data access between one application or service and another. In web programming, an API serces as a means for website to exchange data with an online service. The API provides service functionality through a URL (Uniform Resource Locator) by sending requests via HTTP (Hypertext Transfer Protocol) and return responses in either XML (Extensible Markup Language) or JSON (JavaScript Object Notation) format.



Picture 1. how api work

* + 1. **Django Framework**



Picture 2. Django logo

Django is a framework open-source written with python, it was use to simplify the process of developing complicated websites. Made by two programmer Adrian Holocaty and Simon Willison in 2002. Nowaday django is maintained by the Django Software Foundation (DSF), an indepentdent organization established in the US as a 501©(3) non-profit.

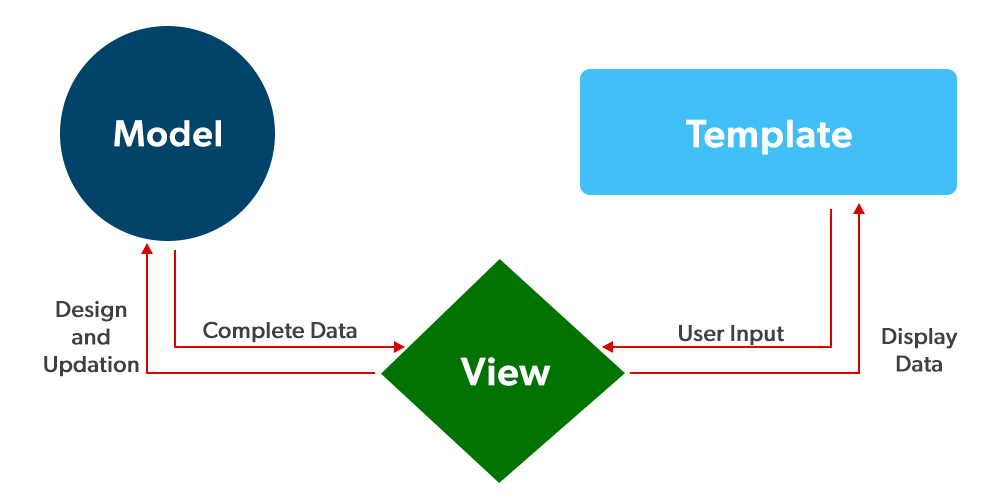
The main goal of django is to aims for component reusability, minimal code, and especially rapid application development. Instead of directly interacting with the database, through Django, we only need to declare data models using Python classes. In this way, tasks like creating, retrieving, updating, and deleting data (CRUD) are automatically supported by Django by sending HTTP requests to the URLs provided by the Django Server.

When using django, it will make the process of creating a website become simple, secure and faster:

* Simple: no database interaction. Can easily change to another database.
* Secure: django provides a user authentication system, account management, and secure password handling.
* Fast: django is make to help developer create a website as fast as possible.

Django is based on MVT (Model-View-Template) architecure which has the following three parts:

* Model: the model is the interface of your data. It responsible for maintaining data. It create database tables base on the classes that you declare in on to connected database (MySql, Postgres)
* View: The view is the user interface that you see in your browser when you render the website. It send data for HTML/CSS/Javascript to process and show the user information. It also where you interact with database.
* Template: A template consist of static parts of the desired HTML as well as some special syntax describing how synamic content will be inserted.



Picture 3. how django work

* + 1. **Python**

Python is a versatile, powerful, and flexible programming language that can be used to develop various types of applications. It is used in many fields, ranging from web development to scientific and engineering applications.

Python plays a crucial role in django as it include many features of Python to provide functionalities and benefirs for developers. Python have serveral advantages, including ease to lear, large community and flexibility.

## **CHAPTER 2: DESIGN AND IMPLEMENTATION**

* 1. **Overview of the website**

The website supports users for managing their incomes, expenses and simulate stock trading. User can register an account, login logout. The system will make the user password secure by encoding user’s password. When user login, user can use functions such as: add, edit, delete expense, add, edit, delete incomes, see stock price in real time and can simulate stock trading, edit date and buy quantity. The website will be implemented based on “Guest”, “User” and “Admin”

When “Guest” access the website, they will be redirect to login screen, “Guest” can only using website after register an account.

When “Guest” registing an account, they has to provide a validate email address, if not it will gray out the register button and cant register. If user name is duplicated, it will show a message that name is taken and the register button will gray out. After making an account, a message for account created will pop up, “Guest” can now go to login screen and then signin their previous registed account, after that, “Guest” will become “User”. In the main menu, “User” can add an expense, an expense will have 4 field amount, description, category, date. After adding expense “User” can edit, delete if wanted. Income is the same at adding expense but with category change to source. In both expenses and incomes have a analysis system that calculate the total price of expenses and total of income, in the income tab it have calculate your actual income after tax and expenses. Both expenses and incomes have export function, which can be exported to csv, excel, pdf. User can also see their incomes and expenses in chart. In the stock tab, User can see stock price in real-time, after that if they want to simulate trading stock, they can head to simulate and try buying some stock, in the stock buying interface, you can choose stock using symbol that you see in the stock tab, after that you can enter quantity and date, the system will automatically get the price of that stock in that day. If the stock symbol is wrong, a message will pop up.

“Admin” will have access to all income, expenses and simulator. Admin can add more category and source for user to choose.

* 1. **Main Functions**

|  |  |  |
| --- | --- | --- |
| **No** | **Actor** | **Functions** |
| 1 | Guest | * Signup * Signin |
| 2 | User | * Signin * Update expenses * Update income * Search through expenses * Search through income * Watch expenses in chart * Watch incomes in chart * Watch live stock price, chart * Update stock simulate purchase |
| 3 | Admin | * Managing expenses * Managing incomes * Mangaing stock purchases |

Table 1. Main functions of the system

* 1. **Use Cases Diagrams**
     1. **Guest**

A diagram of a person with blue circles and text

Description automatically generated

Picture 4. Use case diagram of guest

“Guest” cant actually using the website without signin, the main function of “Guest” is only signup and signin.

* + 1. **User**

A diagram of a flowchart

Description automatically generated

Picture 5. Use case diagram of user

After signing up, “Guest” will become “User”, now user can actually using the website, here some function that user now have: incomes management, expenses management, stock trading simulation.

* + 1. **Admin**

A diagram of a signin

Description automatically generated

Picture 6. Use case diagram of admin

“Admin” can see all the data on the website. There can be only one “admin” which is superuser created through django command. “Admin” can also changing data or adding more data to the database.

* 1. **Database Design**

In django, models are used to define the data structure of a website and create tables in the database. Each model represents a table and defines the fields with their data types.

A screenshot of a computer screen

Description automatically generated

Picture 7. Database design

From here we can use our designed database and transfer it in to database tables using models.py file.

A computer screen with text and a black background

Description automatically generated with medium confidence

Picture 8. Create expense table using models

For example in this Expense table, we don’t need to create id field since django automatic create it. It is the same for other tables.

List of models used:

|  |  |
| --- | --- |
| **Model Name** | **Description** |
| User | Manage account of user |
| Expense | Manage expense of user |
| UserIncome | Manage income of user |
| UsePreference | Manage user’s choosen currency |
| StockPurchase | Manage stock purchase of user |

Table 2. list of installed models

* 1. **Detail Of Functions In The System**
     1. **Authentication**

**Sign up function:**

A diagram of a computer program

Description automatically generated

Picture 9. Flowchart of sign up function

When user sign up, they have to enter their real valid username, email address and password. When user fullfill all the condition, user can now click the register button to create account. If there is any condition is not meet, the system will display an error base on what the condition is and redirect back to register page and use the stored field that user enter before and display it.

**Sign in function:**

A diagram of a computer program

Description automatically generated

Picture 10. Flowchart of sign in function

After created an account, user can login via login page.If user enter the correct username and password, user will be redirect to the main page. Otherwise it will show an error message and user will be redirect back to the login page, the stored username that user had enter before will still be display.

* + 1. **Api Request**

A diagram of a flowchart

Description automatically generated

Picture 11. api flowchart

After sending request through url, it will check is the request is valid. If the request is not valid, it will return 400: bad request. If the request is valid, it will try to query stock data, if it found details, it will return data and end process. Otherwise it will return 404: not found.

* + 1. **User Interface**

1. **Login page**

A white sheet with blue lines

Description automatically generated

Picture 12. login interface

When user go to the website, if user is not logged in, they will be redirect to the login screen. In here user can enter their username and password, if not they can go to the register page to make an account.

1. **Register page**

A screenshot of a computer

Description automatically generated

Picture 13. sign up interface

This is where user will create their account.

A screenshot of a computer

Description automatically generated

Picture 14. error message with taken username

The system will check if username is taken or not, if the username is taken, it will display an error.

A screenshot of a computer

Description automatically generated

Picture 15. error message with take email

The system will check if email is taken or not, if the email is taken, it will display an error.

A screenshot of a register

Description automatically generated

Picture 16. account created when all condition meet

1. **Side bar**

A screenshot of a computer

Description automatically generated

Picture 17. side bar of the website

Side bar on the left of the website is using to navigate through all the different function in the web.

1. **Expense page**

A screenshot of a computer

Description automatically generated

Picture 18. Expense interface

The main interface of expense page, it have all the function for user to manage their own expense and show a total number of expenses.

A white paper with black lines

Description automatically generated

Picture 19. adding an expense

When user want to add an expense, user can press the add expense button and they will be redirect to add-expense page, in there user can choose the amount, description, category and date of expense. After that user can press the submit button and then the expense will be added. If some of the field is empty, user will be redirect back to add expense with an error message.

A white rectangular object with a black line

Description automatically generated

Picture 20. edit or delete an expense

When user want to edit or delete an expense, they can press the edit button and then they can edit amount, description, catetory and date. When user want to delete an expense, they can press the delete button and then the expense will be deleted.

A list of expenses with text

Description automatically generated

Picture 21. export to pdf file

User can export PDF file of your expenses list when press the export PDF button. The PDF file contain all expense’s amount, category, description, date.

A screenshot of a computer

Description automatically generated

Picture 22. export to excel file

User can export to Excel file when press the export excel button. The excel file will have all the field of expense with it data.

A black background with white text

Description automatically generated

Picture 23. export to csv file

User can export to CSV file when press the export csv button. The csv will contain all the data of expense.

1. **Income page**

A screenshot of a computer

Description automatically generated

Picture 24. Income interface

The main interface of income page, it have all the function for user to manage their own income. It also display total income, calculate tax base on Viet Nam 2023’s tax data and show left over after subtract tax and expenses.

A screenshot of a computer

Description automatically generated

Picture 25. add income

When user want to add an income, user can press the add income button and they will be redirect to add-income page, in there user can choose the amount, description, source and date of income. After that user can press the submit button and then the income will be added. If some of the field is empty, user will be redirect back to add income with an error message.

A screenshot of a phone

Description automatically generated

Picture 26. edit or delete income

When user want to edit or delete an income, they can press the edit button and then they can edit amount, description, source and date. When user want to delete an income, they can press the delete button and then the income will be deleted

A close-up of a document

Description automatically generated

Picture 27. export to PDF file

User can export PDF file of your incomes list when press the export PDF button. The PDF file contain all income’s amount, category, source, date.

A table with black text

Description automatically generated

Picture 28. export to excel file

User can export to Excel file when press the export excel button. The excel file will have all the field of income with it data.

A screen shot of a computer

Description automatically generated

Picture 29. export to csv file

User can export to CSV file when press the export csv button. The csv will contain all the data of incomes.

1. **Expenses Summary**

A colorful circle with text

Description automatically generated

Picture 30. chart of expenses

User can get a summary of expense through a donut chart, it have all the expense of user, user can also click on the box to remove that field in the chart.

1. **Income Summary**

A pie chart with different colored circles

Description automatically generated

Picture 31.chart of incomes

User can get a summary of income through a donut chart, it have all the income of user, user can also click on the box to remove that field in the chart.

1. **Stock price view**

A graph of stock market

Description automatically generated

Picture 32. stock price view

In the stock price page, user can see all the stock price, user can change time per block, change stock symbol, …

1. **Stock Trading Simulator**

A screenshot of a computer

Description automatically generated

Picture 33. stock simulator interface

In the stock trading simulator page, user can see all the stock that user has buy. The system will automatic calculate the gain lost percent base on the price of stocks in that day. It also calculate total purchase and total stock value right now.

A white background with a black line

Description automatically generated

Picture 34. buying stock interface

When user want to buy stock, user can press the “simulate another purchase”, after that user will be redirect to buying stock interface. User can enter a stock symbol with it quantity, and the date that user want to buy, the system will automatic get the price of that stock in the choosen date, and add the price for that stock.

A white background with black and white text

Description automatically generated with medium confidence

Picture 35. edit or delete bought stock

User can edit or delete stock when press on the edit button, after that user will be redirect to edit bought stock interface. In edit buy interface, user can change the quantity and purchase date, the system will automatic change the price base on the date of user input date.

## **CHAPTER 3: TESTING AND EVALUATION**

* 1. **Target of testing**

The target of this testing is to check if the website is working correctly with all theprogrammed function, if there is any bug that discover through the testing, it can be used for further optimize the website in the future.

* 1. **Testing senario**

**Senario 1: Sign up, sign in**

Start testing the sign up, sign in function:

* Sign in wrong username or password
* Sign in right username or password
* Sign up with duplicated username, email
* Sign up with password less than 6 character
* Sign up with validate username, email, password
* Try go to main page url without login
* Try go to main page after login
* Sign out

**Senario 2: Expense, Income management**

Start testing expense, income management function:

* Try go to expense main page
* Try adding an expense with empty field
* Try adding an expense with valid field
* Try editing an expense with empty field
* Try editting an expense with valid field
* Try delete an expense
* Try go to expense chart
* Try remove some expense field in chart
* Try go to income main page
* Try adding an income with empty field
* Try adding an income with valid field
* Try editing an income with empty field
* Try editting an income with valid field
* Try delete an income
* Try go toi income chart
* Try remove some income field in chart

**Senario 3: Stock trading simulator**

Start testing stock trading function:

* Try go to stock price view
* Try changing stock symbol
* Try go to stock trading simulator page
* Try buying a stock with empty field
* Try buying a stock with valid field
* Try edit a stock with empty field
* Try edi a stock with valid field
* Try delete a stock

**Testing enviroment:**

* Testing device: Laptop Lenovo Legion 5
* Device configuration:
* Cpu: Intel Core i7 12700H
* Ram: 16 GB
* Windows: 11
* Browser: google chrome
  1. **Result of testing**

Result after testing:

|  |  |
| --- | --- |
| **Senario** | **Result** |
| Senario 1 | Pass |
| Senario 2 | Pass |
| Sernario 3 | Pass |

Table 3. Result of testing

**PART 3 : CONCLUSION AND FUTURE WORK**

1. **Conclusion**

Finance management website has readed it goal. The website provide a friendly UI/UX with working function like manage expense, income and simulate stock trading. We hope that the website will prove it usefulness for student or any individual that need to manage their own finance.

However, the website still missing some function like bill reminders, debt management, analysis user behavior. The process of adding expense and income need to much action. Also due to using api, some time the time to load stock simulate is longer than usual.

1. **Future Work**

In the future, we plan to improve the interface of expense and income page, making it easier to watch all the data and make it easier to add data with repeatable item can be add with one button. Next we will find a way to make request to api faster by only need to request one time for the same symbol. Finally we will try to make more functions for user, further improve the user experiment with our website.

**REFERENCES**

1. Tạp chí công thương for stock market news,

<https://tapchicongthuong.com.vn/bai-viet/so-tai-khoan-chung-khoan-tai-viet-nam-da-tuong-duong-hon-8-dan-so-111769.html>

1. Django framework using for making the website,

<https://www.djangoproject.com/>

1. Yahoo finance api for getting stock price in the past,

<https://python-yahoofinance.readthedocs.io/en/latest/api.html>

1. Boostrap for premake html,

<https://getbootstrap.com/>

1. Postgresql using for database of django,

<https://www.postgresql.org/>

1. Finnhub api using for getting stock price at current time,

<https://finnhub.io/docs/api>

1. Chartjs for premake chart,

<https://www.chartjs.org/>