

**A PROJECT REPORT
For
Mini Project (KCA353)**

**Session (2023-24)
Submitted by
Aayushi Singh
2200290140003
Ankita Pandey
2200290140030**

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
Mr. Praveen Kumar Gupta
Assistant professor**



**Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206**

(FEBRUARY 2024)

DECLARATION

I hereby declare that the work presented in this report entitled “Expense Tracker System”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Aayushi Singh (2200290140003)

Ankita Pandey (2200290140030)

Branch: MCA

CERTIFICATE

Certified that **Aayushi Singh 2200290140003, Ankita Pandey 2200290140030** has/ have carried out the project work having “**Expense Tracker System**” (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:04/01/2024

Aayushi Singh 2200290140003

Ankita Pandey 2200290140030

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:04/01/2024

Mr. Praveen Kumar Gupta
Assistant professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head of Department
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

In modern economic times, everyone wants to save their money and keep expenses within their budget limit. But, due to our busy schedules in our day to day life, we may lose track of our expenses and end up overspending which leads to debts. In this project, the aim is to develop a mobile application which helps user to keep track of all the expenses and to simplify the tracking process.

The proposed system will also generate detailed information on which category of shopping we are exceeding our budgets so that the user can monitor and keep his expenses within his budget limit. The rapid growth in technology has led to smart mobile applications development.

Budgeting on a personal level is always a problematic issue faced on a daily basis by almost each of us. Recording and managing expenses, income/savings are getting very difficult in today's busy life which ultimately results in the inefficient planning of personal budget.

This proposed application will allow users to track and analyse their expenses, income, and savings to plan their personal budget, organize their expenses & income and record movement of money. The application will allow users to set their monthly budget (overall budget or by expense category) to monitor and control their spending with the help of budget management.

The application will remind the user for any pending bills, expense limit exceeding mentioned in a budget well before time. The application will provide different reports regarding expenses and income on the daily, weekly, monthly or yearly basis for better control over the money.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Mr. Praveen Kumar Gupta** for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Aayushi Singh

Ankita Pandey

Table of Content

Content	Page Number
Declaration	ii
Certificate	iii
Abstract	iv
Acknowledgement	v
Chapter 1 - Introduction	1-2
1.1 Motivation	1
1.2 Expected Outcome	1
1.3 Comparative Studies	2
1.4 Chapter Overview	2
Chapter 2- Literature Review	3-4
2.1 Introduction	3
2.2 Literature Survey	3
2.3 Node.js Platform	4
Chapter 3- Requirement Analysis and Feasibility Analysis	5-19
3.1 Data Collection Method	5
3.1.1 Source of Data	5
3.2 Requirement Specification	5
3.2.1 Functional Requirement	5-6
3.2.2 Flow Chart	7-8
3.2.3 Use Case Diagram	9
3.2.4 Non-Functional Requirement	10
3.3 Feasibility Analysis	10
3.3.1 Technical Feasibility	10
3.3.2 Operational Feasibility	11
3.3.3 Economical Feasibility	11-12

3.3.4 Constraints	12
3.4 Structure System Requirement	13
3.4.1 ER Diagram	13
3.4.1.1 ER Notation	13
3.4.1.2 Data Modelling(ER Diagram)	14
3.4.2 Data Flow Diagram	15
3.4.2.1 Process Modelling (DFD-0)	16
3.4.2.2 Process Modelling (DFD-1)	17
3.4.3 Activity Diagram	18-19
Chapter 4- Design	20-33
4.1 Dashboard Panel	20-21
4.1.1 Graph Representation	22
4.2 View Transaction Module	23
4.2.1 Date Wise Graph Representation	24-25
4.3 Income Module	26-27
4.3.1 Add Title of Income	28
4.3.2 Add Amount of Income	29
4.3.3 Add Date of Income	30
4.4 Expense Module	31
4.3.1 Add Title of Expense	32
4.3.2 Add Amount and Date of Expense	33
Chapter 5- Testing	34-38
5.1 Unit Testing	34
5.2 Module Testing	35
5.3 Sub-system Testing	36
5.4 System Testing	37
5.5 Acceptance Testing	38

Conclusions	39
References	40
Bibliography	41

CHAPTER 1

INTRODUCTION

A daily expense tracker is a one kind of digital diary that helps to keep an eye on all of our money related transitions and also provides all financial activities report daily, weekly, monthly and yearly. Users get notification to record expenses and incomes that are helpful to the tracking system of the application. All information is saved in offline mode so users can easily access any time and any palaces. User interface of the daily expense tracker is very simple and attractive so it is easy to understand and the best way to record our financial data.

1.1 Motivation

The motivation to work in this project is actually our real-life experience. As a user We face many difficulties in our daily file. In our daily life money is the most important portion and without it we cannot last one day on earth but if we keep on track all financial data then we can overcome this problem. Most of the people cannot track their expenses and income one way they face the money crisis and depression. This situation motivates us to make an android app to track all financial activities. Using the Daily Expense Tracker user can be tracking expenses day to day and making life tension free.

1.2 Expected Outcome

The main objective of this project is support to the user to sustain all financial activities like digital automated dairy. This application helps the user to avoid unexpected expenses and bad financial situations.

Using this application, users can manage all financial data and track all expense and income category wise.

► Creating a category and recording all expenses and income under the category.

- Enable the notification system user get notification daily at a specific time that can help the user insert expense and income.
- Backup and Restore all information.
- Report are generated in PDF format in category wise or time period.

1.3 Comparative Studies

We have studied some similar applications and found some problems that are not working in offline mode. There are some limitations to their application, which we do not have in our application. In some applications, there are no login and signup options, which is required for the security of a user's information. However, in our application, we have a login and signup which is required for user data security and has a dashboard for monitoring the entire system. This application is a very simple and user-friendly application for the common people. The main goal of the project is to make the system offline and perform more tasks in a short period of time.

1.4 Chapter Overviews

Chapter-2- Literature Review- This chapter will contain data regarding introduction of our app, theory behind the app making. It will also contain detailed analysis of how we searched and collected ideas for the same. Then it will include the conclusion of a literature review of how we came to the conclusion of deciding our work.

Chapter-3- Requirement Analysis and Feasibility Analysis - In Chapter 3 we discussed about the Data collection method , Functional and Non-Functional Requirement , Feasibility Analysis ,Data modelling and Process Modelling.

Chapter-4-Design- This will include the screenshots of the modules.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A writing audit is a study of insightful sources on a particular research. We found various similar products that have already been developed in the market. Unlike all those products, Personal Expense Tracker (PET) provides security and graphical results. We provide the users to enter their wish-list before any purchase. It generates notifications to notify users about their timely entry. In order to complete our task, we used the Android platform to build a portable, handy product that can run on an Android phone. We used Android studio to build our application. The major Language we used for scripting was JavaScript , HTML , CSS and NodeJS for producing better layout. We used MongoDB to implement the database.

2.2 Literature Survey

The literature survey for the Expense Tracker System project reveals a substantial body of research and development in the field of financial management and budget tracking. Numerous studies emphasize the significance of effective expense tracking for individuals, businesses, and organizations to achieve fiscal responsibility and sustainable financial health. Existing systems and applications showcase diverse approaches, incorporating features such as real-time data synchronization, intuitive user interfaces, and advanced analytics. Additionally, research highlights the importance of mobile accessibility and cloud integration in contemporary expense tracking solutions. Scholars underscore the need for security measures to safeguard sensitive financial information, driving the integration of encryption and authentication mechanisms in such systems. The literature survey informs the current project by drawing insights from these studies to enhance the robustness, user-friendliness, and security of the proposed Expense Tracker System.

2.3 Node.js Platform

In the context of an Expense Tracker System project, Node.js is a powerful platform for building the server-side component of the application. Node.js is known for its event driven, non-blocking I/O model, which makes it well-suited for handling concurrent operations, such as managing multiple user requests in real-time.

By leveraging Node.js, developers can create a scalable and efficient backend for the Expense Tracker System. Node.js can handle asynchronous tasks, making it suitable for dealing with database operations, user authentication, and communication with external APIs, all of which are crucial functionalities in an expense tracking application. Additionally, the extensive npm (Node Package Manager) ecosystem provides a wealth of modules and libraries that can be used to accelerate development and enhance the system's features. Furthermore, Node.js is often paired with frameworks like Express.js to simplify the process of building RESTful APIs, routing, and handling HTTP requests. This combination can facilitate the creation of a responsive and performant backend for the Expense Tracker System, allowing for smooth communication between the client-side and the server-side components of the application.

CHAPTER 3

REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

3.1 Data Collection Methods

3.1.1 Source of Data

Interview: Interviews were carried out with some of the students in our own college asking about the expenses that they do in day to day life. While taking those samples we got that they always broke off at the end of the month, which means they do not end up calculating those expenses that they spend day to day. So in order to control the unnecessary spending habits expense tracker is must. While using this tracker they can control their expenses and also save some of those too.

I.Questionnaire: Set of questionnaire were prepared to gain knowledge about how people track their budget. This process conclude that maximum of them do not plan for what they have earned and no track at all.

3.2 Requirement Specification

3.2.1 Functional Requirements

1. Dashboard panel

The system shall authenticate the user and then display panel based on the particular identified user.

2. Add bill

The system shall allow the user to add bill details based on the user's need to track the type of expenses.

3. Expense planner

The system should graphically represent the current month figure based on user's current month expenses and user's own budget share.

4. Expense tracker

The system should graphically represent the yearly expense numbers in form of report.

5. Add References

The system shall allow users to add notes to their expenses and income.

6. Calander

The system shall allow users to add the date to their expenses and income.

7. Category

The system shall allow users to add categories of their expenses and income.

3.2.2 Flow Chart

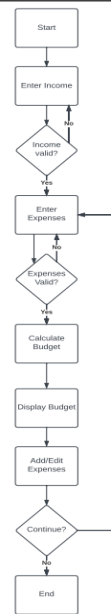


Fig 1: Flow Chart

Here is a possible flowchart for a basic expense tracker system based on the given information:

1. **Start:** The budgeting app starts by asking the user to enter their income.
2. **Enter Income:** The user enters their income into the app.
3. **Income valid?:** The app checks whether the income entered by the user is valid. If the income is not valid, the app will ask the user to enter it again.
4. **Enter Expenses:** Once the income is valid, the app asks the user to enter their expenses.
5. **Expenses Valid?:** The app checks whether the expenses entered by the user are valid. If the expenses are not valid, the app will ask the user to enter them again.
6. **Calculate Budget:** Once the income and expenses are valid, the app calculates the budget by subtracting the expenses from the income.
7. **Display Budget:** The app displays the calculated budget to the user.
8. **Add/Edit Expenses:** The user has the option to add or edit their expenses.

9. **PRE:** This step is not clear from the flowchart, but it could mean that the app is checking whether there are any pending tasks or updates.
10. **Continue?:** The app asks the user whether they want to continue using the app.
11. **End:** If the user chooses to end the session, the app will close

3.2.3 Use Case Diagram

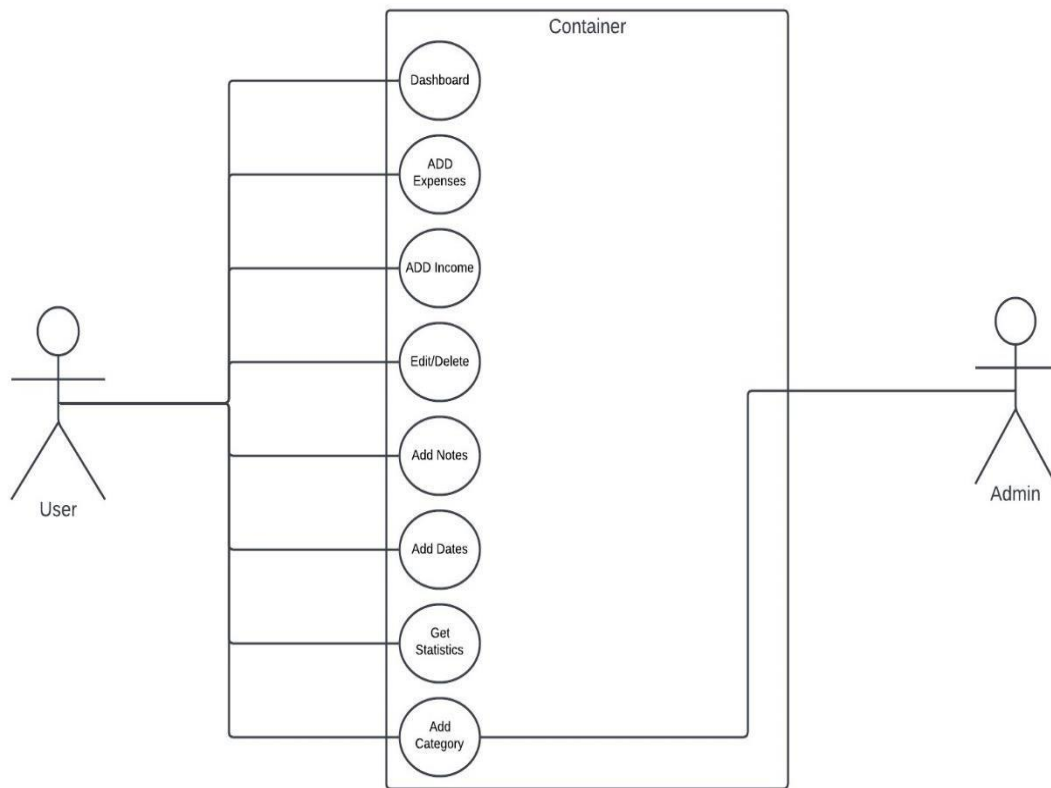


Fig 2: Use Case Diagram

The given code is a user interface (UI) for a personal finance app. It is written in a markup language called HTML and uses a CSS framework called Bootstrap for styling. The UI consists of several sections:

Dashboard: This is the main section where the user can view their total income, expenses, and net balance.

Expenses: This section allows the user to add, edit, or delete expenses.

Add Income: This section allows the user to add income.

Add Notes: This section allows the user to add notes or descriptions for their expenses or income.

Add Dates: This section allows the user to add dates for their expenses or income.

Statistics: This section provides the user with various statistics about their expenses and income, such as the average expense or the most common category of expenses.

Category: This section allows the user to categorize their expenses.

Container: This section contains all the other sections and ensures proper spacing and alignment.

Admin: This section is for the admin user, who can manage all the users and their data.

3.2.4 Non-Functional requirements

1. Usability

There is a consistency in all the modules and webpages. To ease the navigation there is a back tab to provide access to previous page. There is proper instruction on each page.

2. Reliability

Each data record is stored on a well-built efficient database schema. There is no risk of data loss. The internal evaluation of data is well coded.

3. Supportability

The system is well built to support any machine. Maintainability of the system is easy.

4. Performance

In order to ease the accessibility, the types of expenses are categorized along with an option to name on the own. Throughput of the system is increased due to light weight database support

5. Availability

The system is available all the time, no time constraint.

3.3 Feasibility Analysis

3.3.1 Technical Feasibility

The technical feasibility of a expense tracker flowchart refers to the app's ability to be developed using current technology and resources. Based on the given flowchart, the expense tracker requires user input for income and expenses, performs calculations to determine the budget, and allows the user to add or edit expenses.

To determine the technical feasibility of this flowchart, several factors must be considered, including:

1. **Data Input:** The app requires user input for income and expenses. Therefore, the app must have a user-friendly interface for data input.
2. **Data Validation:** The app checks whether the income and expenses entered by the user are valid. Therefore, the app must have a mechanism for data validation.
3. **Calculation:** The app calculates the expense by subtracting the expenses from the income. Therefore, the app must have a mechanism for performing mathematical calculations.
4. **Data Display:** The app displays the calculated expenses to the user. Therefore, the app must have a mechanism for displaying data in a user-friendly format.

3.3.2 Operational Feasibility

Operational feasibility in the context of an expense tracker refers to the evaluation of whether the organization is capable of implementing and maintaining the expense tracker. This includes assessing the availability of necessary resources, such as staffing, organizational structure, and any applicable legal requirements. At the end of the operational feasibility study, the team will have a clear understanding of whether they have the resources, skills, and competencies to implement and maintain the expense tracker.

3.3.3 Economic Feasibility

Economic feasibility in an expense tracker refers to the assessment of the financial viability of the product. This includes evaluating the costs associated with developing, marketing, and maintaining the app, as well as estimating potential revenue streams.

When considering the economic feasibility of an expense tracker, it's important to take into account the following factors:

1. **Development and maintenance costs:** This includes the cost of hiring a development team, purchasing necessary resources and technologies, and ongoing maintenance and updates.

2. **Marketing and advertising costs:** Promoting the app to the target market will require a budget for marketing and advertising efforts, such as social media campaigns, influencer partnerships, and paid search ads.
3. **Revenue projections:** Estimating the potential revenue from the app is crucial for determining its economic feasibility. This can be done by analysing the pricing strategy, market size, and growth potential.
4. **Break-even analysis:** This involves calculating the point at which the revenue generated from the app equals or exceeds the costs associated with its development, marketing, and maintenance.

3.3.4. Constraints:

Constraints in an expense tracker refer to any limitations or restrictions that may impact the development, implementation, or maintenance of the app. These constraints can be technical, organizational, or financial in nature.

Some common constraints in an expense tracker include:

1. **Technical constraints:** These can include limitations in the hardware or software used to develop the app, such as compatibility issues, performance limitations, or security concerns.
2. **Organizational constraints:** These can include limitations in the organizational structure, such as a lack of resources, skills, or expertise, or restrictions on the use of certain technologies or practices.
3. **Financial constraints:** These can include budget limitations, revenue projections, or the cost of development, marketing, and maintenance.
4. **Legal constraints:** These can include compliance with laws and regulations, such as data privacy laws, financial reporting requirements, or industry-specific regulations.
5. **Time constraints:** These can include deadlines for development, implementation, or maintenance, or limitations on the amount of time available .

3.4 Structure System Requirement

3.4.1 ER Diagram

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model. It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables. It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user. In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

3.4.1.1 ER Notations:

There is no standard for representing data objects in ER diagrams. Each model methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX. All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- ♣ Entities are represented by labelled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- ♣ Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
- ♣ Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns. Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

♣ Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

3.4.1.2 Data Modelling (ER Diagram)

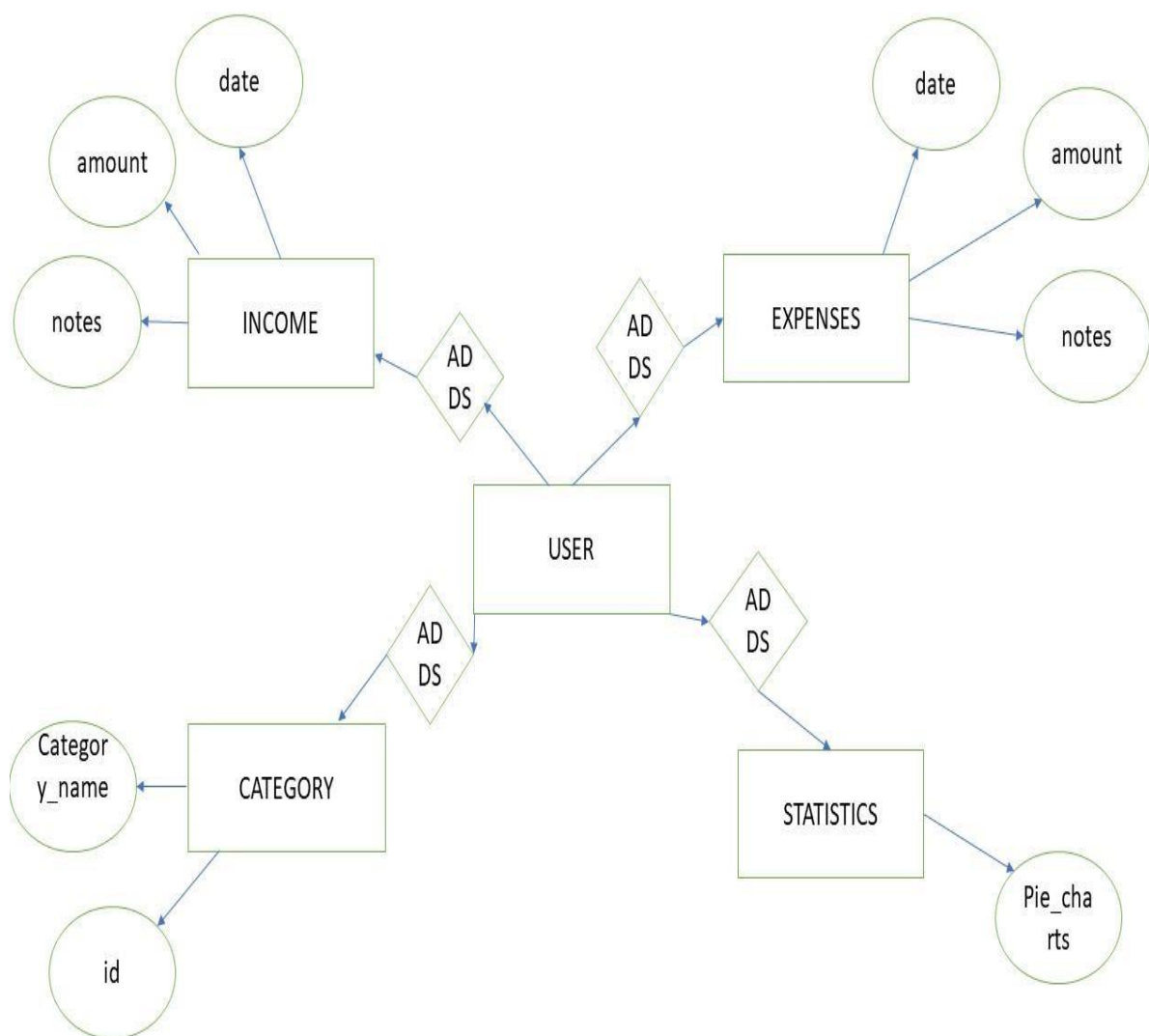


Fig 3: ER Diagram

The above diagram explains the relationship between the databases where rectangle represents entity, oval represents attributes and diamond represents relation. There are four entities with their respective attributes.

3.4.2 Data Flow Diagram:

A data flow diagram is a short road map for that graphically represents how the data moves through the existing system. we have used data flow diagram in design process. The data flow diagram provides facilitating communication between us and user. DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

Here are some common symbols used in data flow diagrams (DFD) and their explanations:

1. **External Entity:** Represents an external system or user that interacts with the system being model. It can be a person, another system, or a hardware device. They are represented by rectangles, circles, or squares, and are usually placed on the diagram's edges.
2. Processes are procedures that manipulate the data and its flow by taking incoming data, changing it, and producing an output. They are represented by rounded rectangles and usually start from the top left of the DFD and finish on the bottom right of the diagram.
3. Data stores hold information for later use, like a file of documents that's waiting to be processed. They are represented by open rectangles.
4. Data flow is the path the system's information takes from external entities through processes and data stores. With arrows and succinct labels, the DFD can show you the direction of the data flow.

3.4.2.1 Process Modelling (DFD-0)

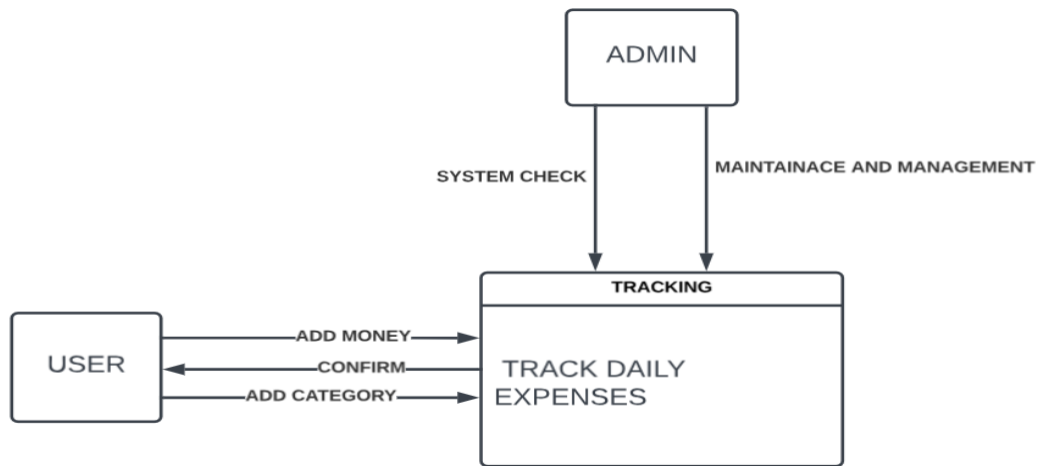


Fig 4: DFD Level 0

The above diagram shows the DFD level-0 where user adds money to the tracking system. The tracking system update it to the database then data base will retrieve it to tracking system. After retrieving the system confirms data to the user. The admin checks system and maintains the system.

3.4.2.2 Process Modelling (DFD-1)

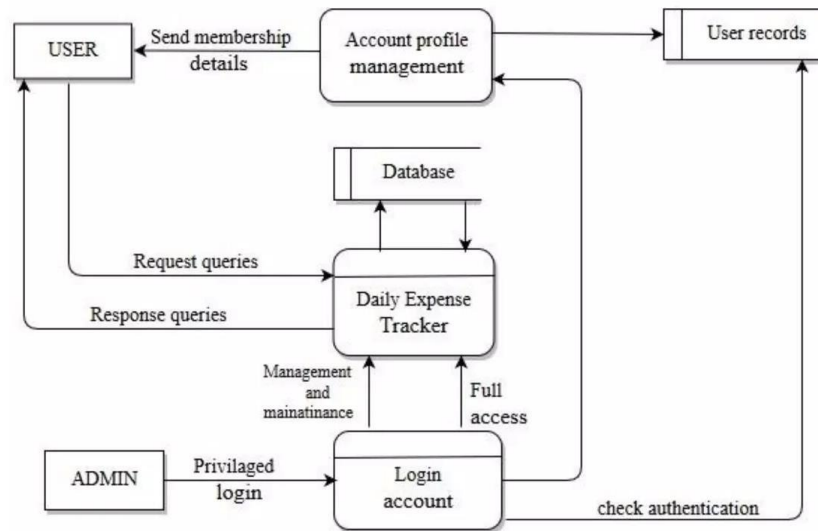


Fig 5: DFD Level 1

The above diagram is for DFD level I where we can track data through the processes it is involving. The process daily expense tracking is further divided as account profile management and login account management. The user requests queries to the daily expense tracker and gets response from it. The login account checks the authentication of the user records. The account profile management sends membership details to the user.

3.4.3 Activity Diagram –

It appears you have provided a list of words and phrases, some of which are commands or options that might be found in the user interface of a budgeting app. Here is an explanation of what each word or phrase might represent in this context:

Pie Chart: A type of data visualization that shows proportions of a whole using a circle divided into slices. In a budgeting app, a pie chart might be used to show the distribution of expenses across different categories.

History: A record of past transactions or activities. In a budgeting app, the history section might allow users to view their past expenses, income, and other financial data.

Show Dashboard: A command that displays the main screen or dashboard of the app, which might provide an overview of the user's current financial status and allow them to access different features of the app.

Budget: A plan for managing and allocating money over a certain period of time. In a budgeting app, the budget feature might allow users to set spending limits for different categories of expenses and track their progress throughout the month.

Add Expenses: A command that allows users to record and categorize new expenses in the app.

Add Category: A command that allows users to create new categories for expenses in the app.

Exit: A command that allows users to exit a particular screen or feature of the app and return to the previous screen or the main dashboard.

Overall, these words and phrases suggest that the user interface includes options for viewing and analysing past financial data, tracking current expenses and budgets, and adding new expenses and categories as needed.

3.4.3.1 Activity Diagram- This is the Activity diagram of Expense Tracker .

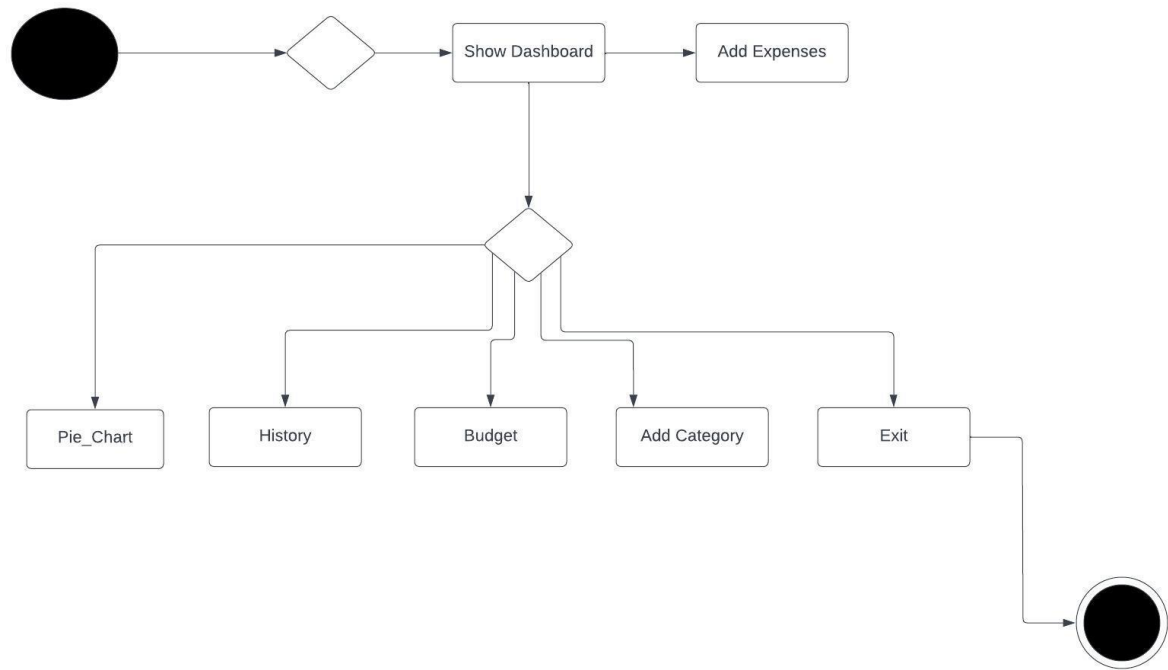
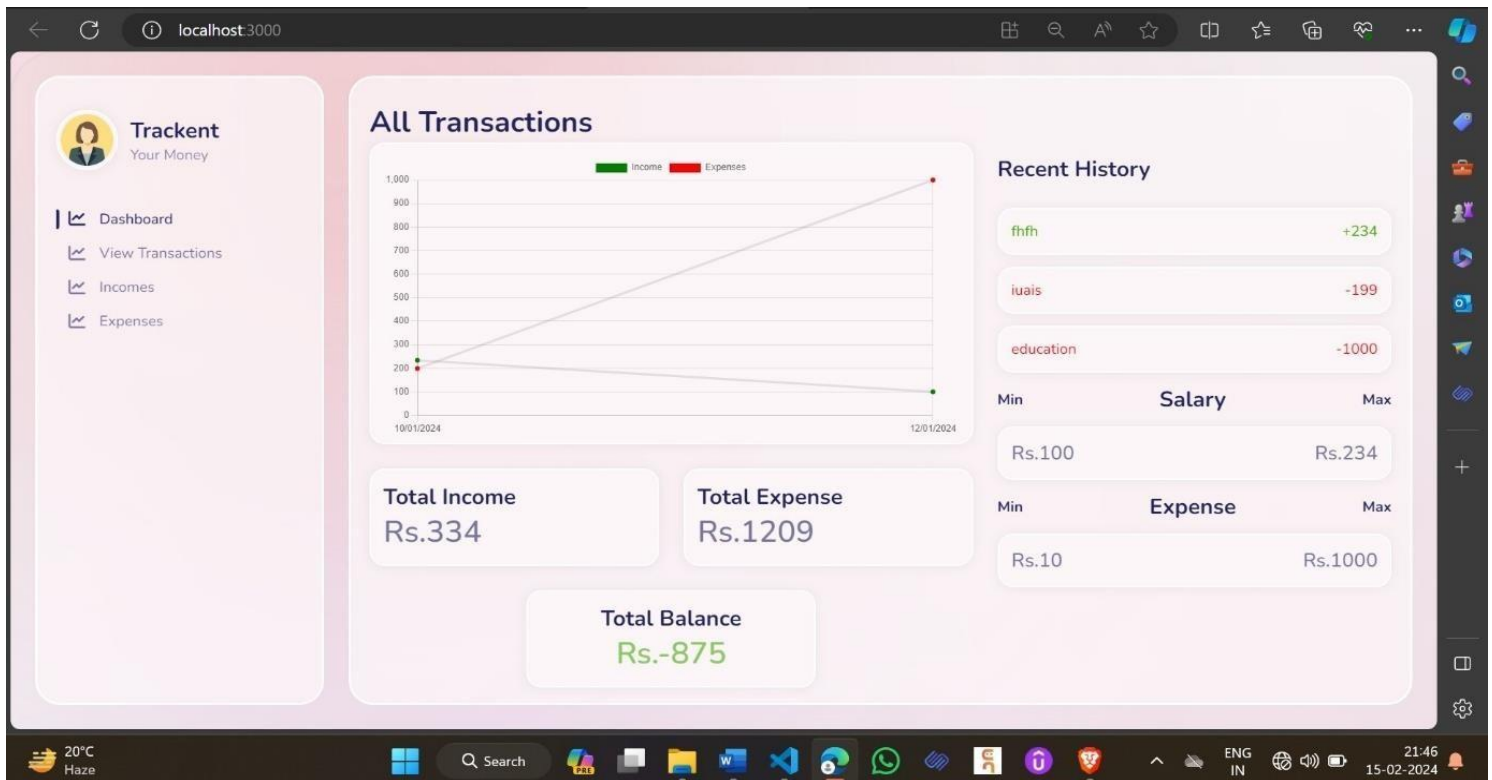


Fig 6: Activity Diagram

CHAPTER 4

DESIGN

4.1 Dashboard Panel



The dashboard module in an expense tracker system project typically serves as the main interface for users to visualize and analyze their financial data. Here's how it generally works:

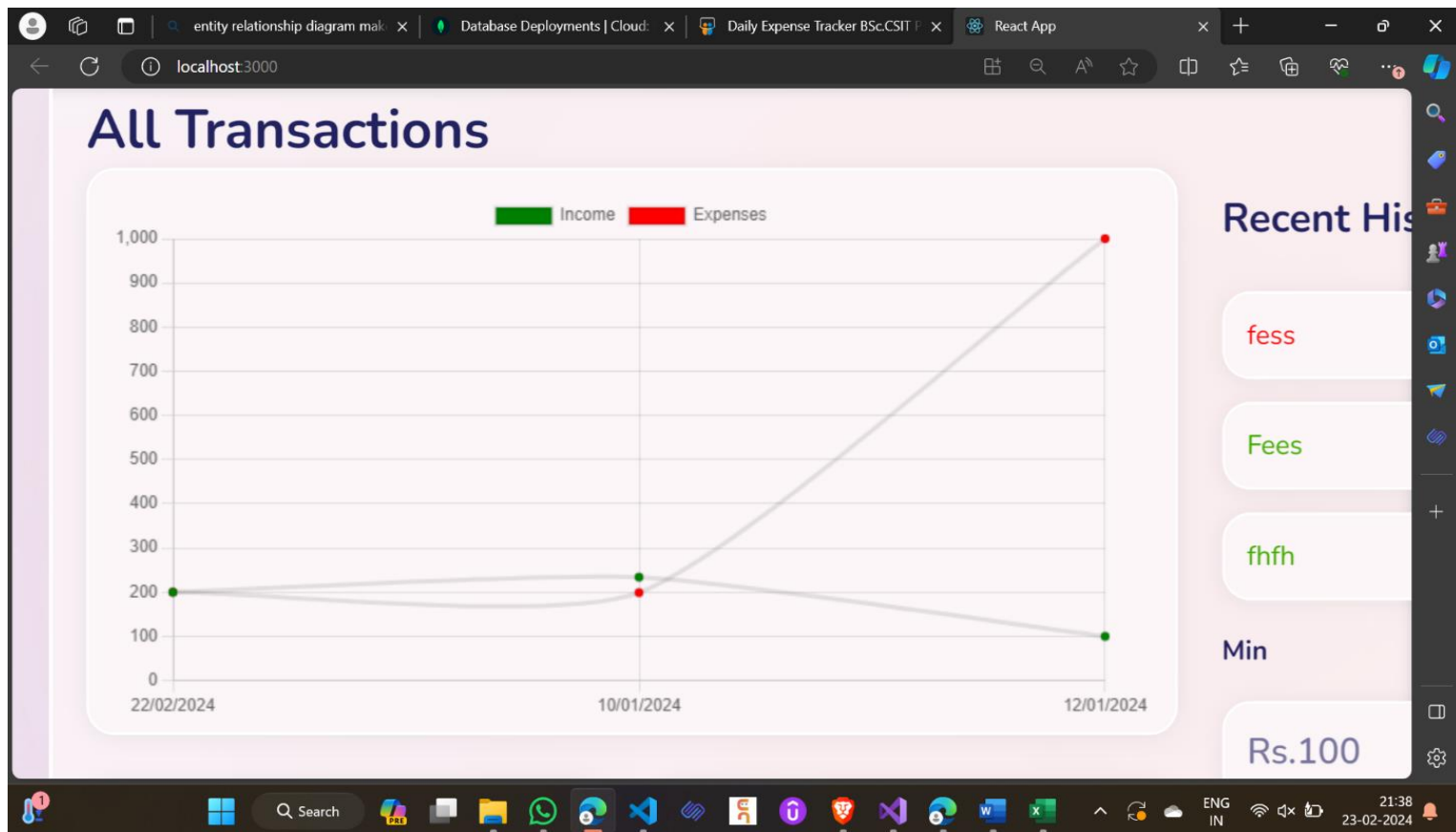
Data Aggregation: The dashboard module collects data from various sources within the expense tracker system. This includes user-entered expenses, income data, budget allocations, and any other relevant financial information.

Data Processing and Analysis: Once the data is collected, the dashboard module processes it to generate meaningful insights. This may involve calculations such as total expenses, income vs. expenses comparison, spending categories breakdown, trends analysis, etc.

Visualization: The processed data is then presented to the user in a visually appealing and easy-to-understand format. This typically includes charts, graphs, tables, and other graphical elements. Common visualizations include pie charts for category breakdowns, line graphs for trends over time, and bar charts for expense comparisons.

Customization and Interaction: Users are often provided with options to customize the dashboard according to their preferences. They may be able to choose which data to display, set date ranges for analysis, adjust visualization settings, and so on. Interaction with the dashboard allows users to drill down into specific data points for more detailed information.

4.1.1 Graph Representation:



This graph appears to represent the income and expenses for a certain time period, with income shown above the x-axis and expenses shown below the x-axis. The x-axis seems to represent time, with specific dates possibly including January 10 and January 12, 2024.

The graph shows a series of vertical bars representing the amount of income or expenses on each date. The highest income bar is on the far right of the graph, with a value of approximately 1,000. There are several other income bars with values of 900, 800, 700, and 600.

On the expense side, there are several bars with values of 100, 200, 300, 400, and 500. The highest expense bar is on January 12, with a value of 800.

Overall, this graph provides a visual representation of the income and expenses for the time period shown, with higher values indicated by taller bars. It can help the user quickly identify trends and patterns in their financial data.

4.2 View Transaction Module



In an expense tracker system, the view transaction module typically refers to the functionality that allows users to see their recorded transactions. Here's how it might work:

Transaction Retrieval: Upon accessing the view transaction module, the system retrieves transaction data associated with the logged-in user. This data usually includes details such as the date of the transaction, the amount spent or earned, category (e.g., food, transportation, entertainment), and any additional notes.

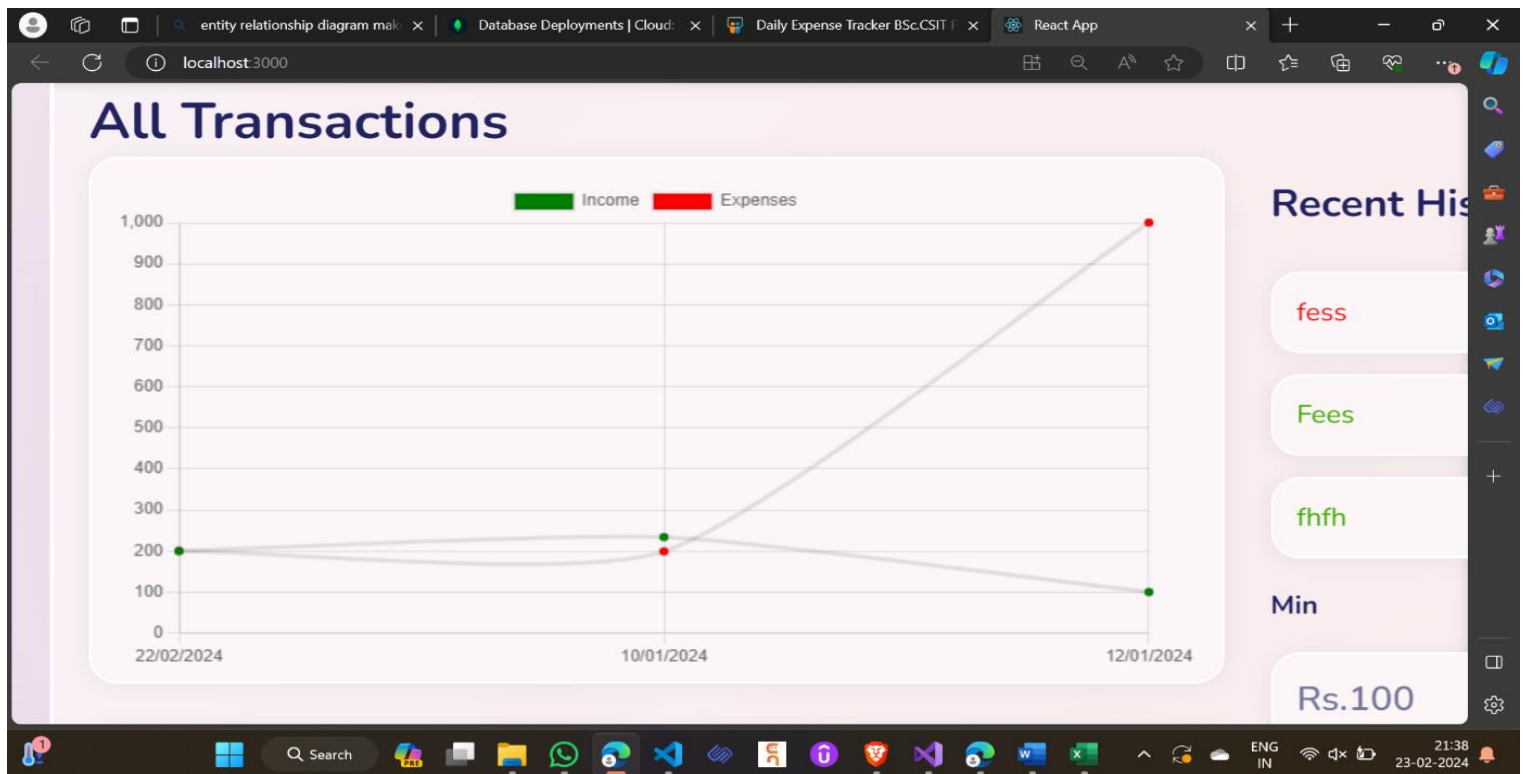
Displaying Transactions: The retrieved transaction data is then displayed to the user in a user-friendly format, such as a list or table. The user may be able to filter transactions based on various criteria, such as date range, category, or transaction type (income or expense).

Sorting and Filtering: Users may have the option to sort transactions by date, amount, or category, and apply filters to view specific subsets of transactions.

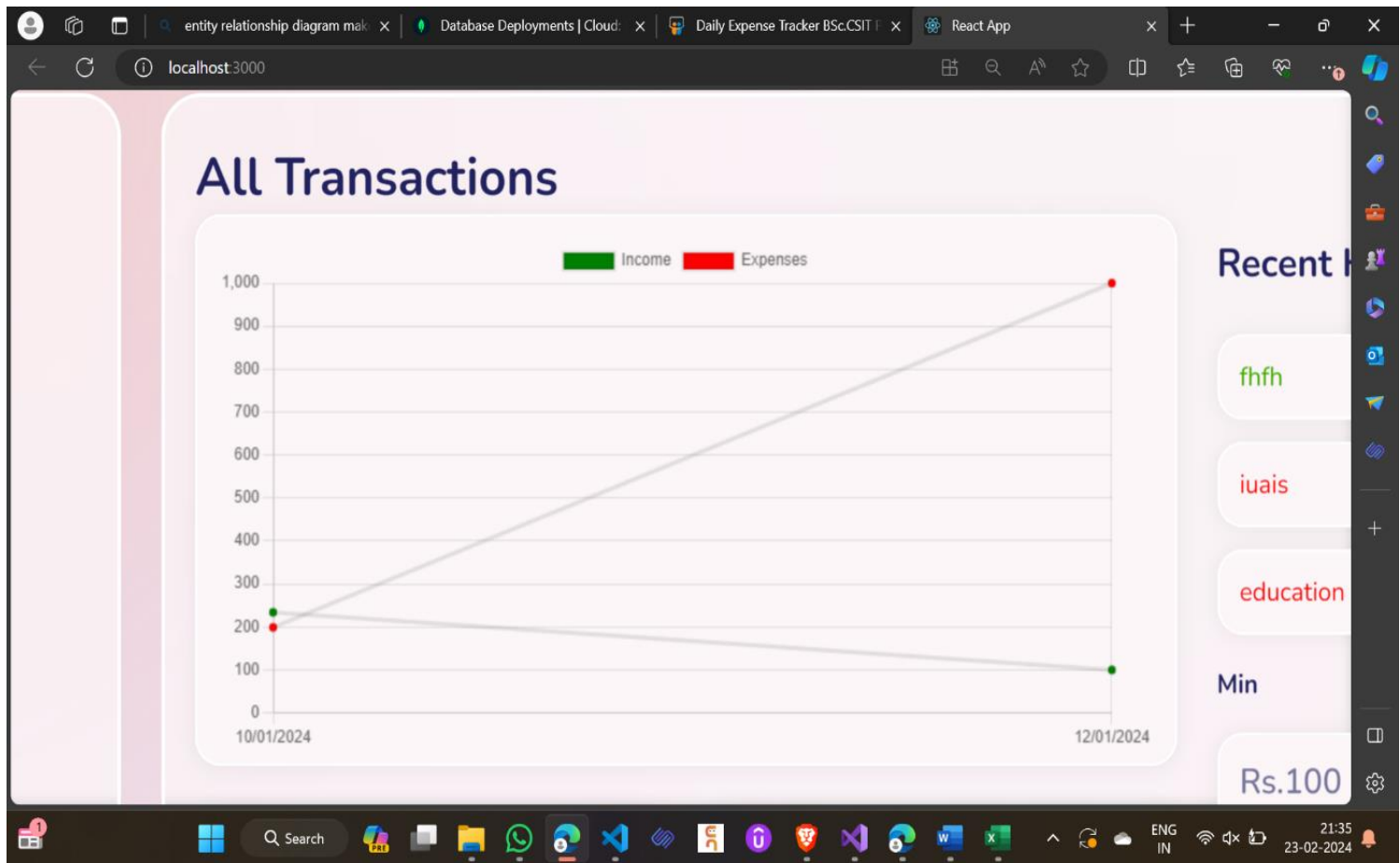
Transaction Details: Users can typically click on individual transactions to view more detailed information, edit transaction details if allowed, or delete transactions if needed.

Visualizations: Some expense tracker systems may offer visual representations of transaction data, such as charts or graphs, to help users analyze their spending patterns over time.

4.2.1 Date Wise Graph Representation:

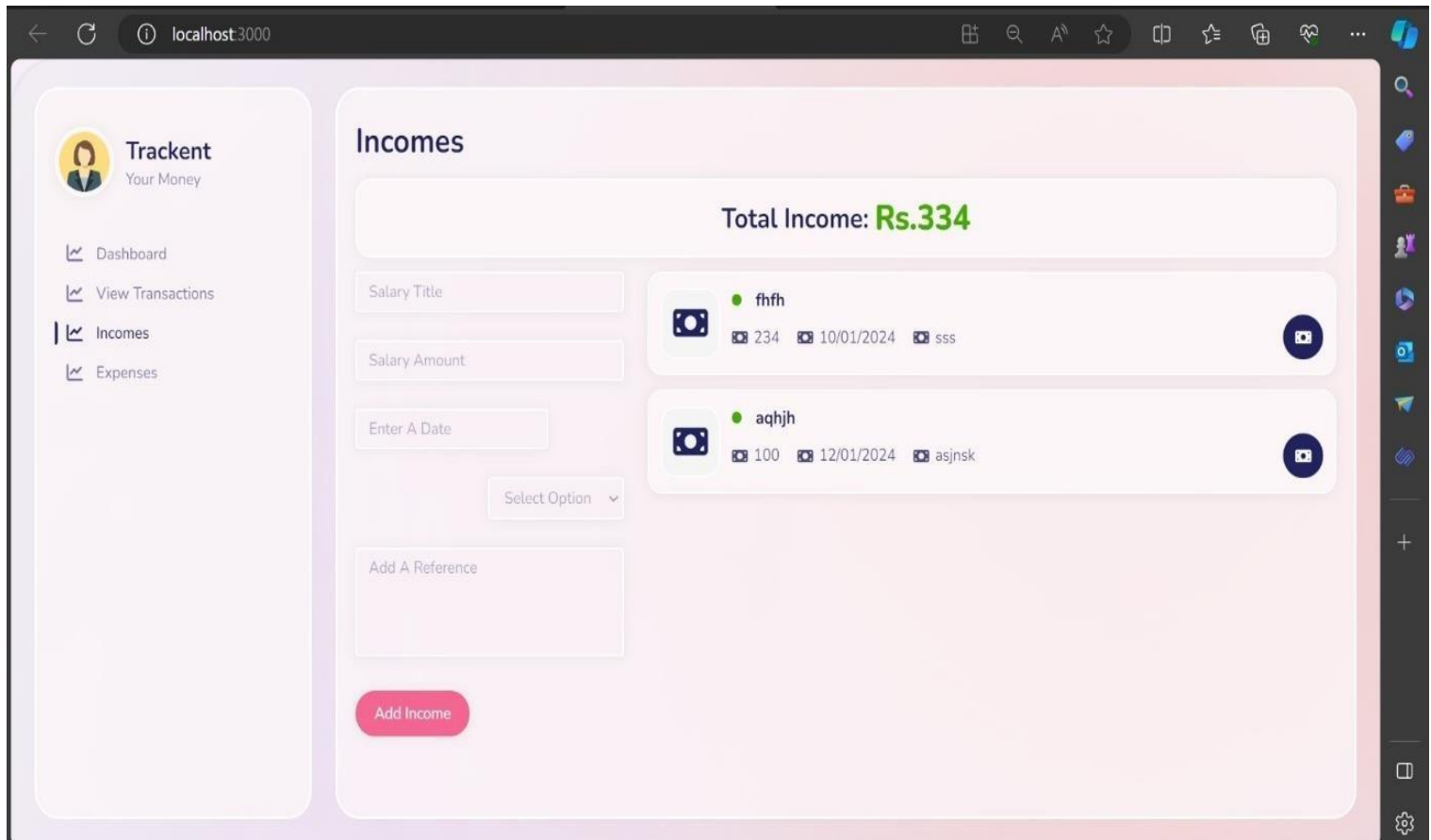


Overall, this financial tracking system displays information about income and expenses, including the amount, date, and category of each transaction. The user can search or filter transactions, and view a summary of the total income and expenses for the time period shown. The system also includes a minimum amount required for a transaction and a current balance or total amount of money being tracked. Additional fields may be available for notes or references.



Overall, this chart displays financial transactions, specifically income and expenses, including the amount and date of each transaction. The user can search or filter transactions, and view a summary of the total income and expenses for the time period shown. The chart also includes a minimum amount required for a transaction. Additional fields may be available for notes or references.

4.3 Income Module



The Income module in an expense tracker system typically functions to help users track and manage their incoming finances, such as salaries, bonuses, investment dividends, etc. Here's how it might work:

Income Entry: Users can manually input details about their sources of income. This includes the amount, the date it was received, the source (e.g., salary from a job, freelance income, etc.), and any additional notes.

Income Categories: Users can categorize their income for better organization and analysis. Common categories might include salary, freelance income, bonuses, investments, gifts, etc. Users might be able to create custom categories to fit their specific needs.

Recurring Income: Users can set up recurring income entries for sources that provide consistent payments on a regular schedule (e.g., monthly salary, quarterly dividends,

etc.). The system can automatically generate these entries based on the specified frequency.

Income Analysis: The system can provide users with insights into their income patterns over time. This might include visualizations like graphs or charts showing income trends, comparisons between different income sources, and forecasts based on historical data.

Budgeting and Planning: Users can use the income data to create budgets and financial plans. The system can help users set income-related goals, track their progress towards those goals, and make adjustments as needed.

Integration with Expense Tracking: The Income module might be integrated with the Expense module to provide users with a comprehensive view of their financial situation. This allows users to see how their income compares to their expenses, identify areas where they can save or cut back, and make informed financial decisions.

4.3.1 Add Title of Income:

The screenshot displays a web application interface for managing income transactions. The main heading is "Incomes". Below it, a summary box shows "Total Income: Rs.334". The form contains several input fields: "Fees", "Salary Amount", "Enter A Date", and "Add A Reference". A "Select Option" dropdown menu is also visible. Two transaction entries are listed, each with a currency icon, a green dot, a label, an amount, a date, and a reference code.

Label	Amount	Date	Reference
fhfh	234	10/01/2024	sss
aqhjh	100	12/01/2024	asjnsk

This income module input form allows the user to enter the amount, date, and category of an income transaction. Additional fields may be available for notes or references.

The fields in the form include:

- "Fees": This field may be a category or tag for the income transaction.
- "entity relationship diagram make x": It is unclear what this field is for, as it does not seem to be related to income tracking.
- "Incomes": This field may be a label or title for the form.
- "Salary Amount": This field is where the user can enter the amount of the income transaction.
- "Enter A Date": This field is where the user can enter the date of the income transaction.
- "Add A Reference": This field may be optional and allow the user to add a reference or note to the income transaction.

4.3.2 Add Amount of Income:

Incomes

Total Income: **Rs.334**

Fees

200

Enter A Date

Select Option ▾

Add A Reference

fhfh
234 10/01/2024 sss

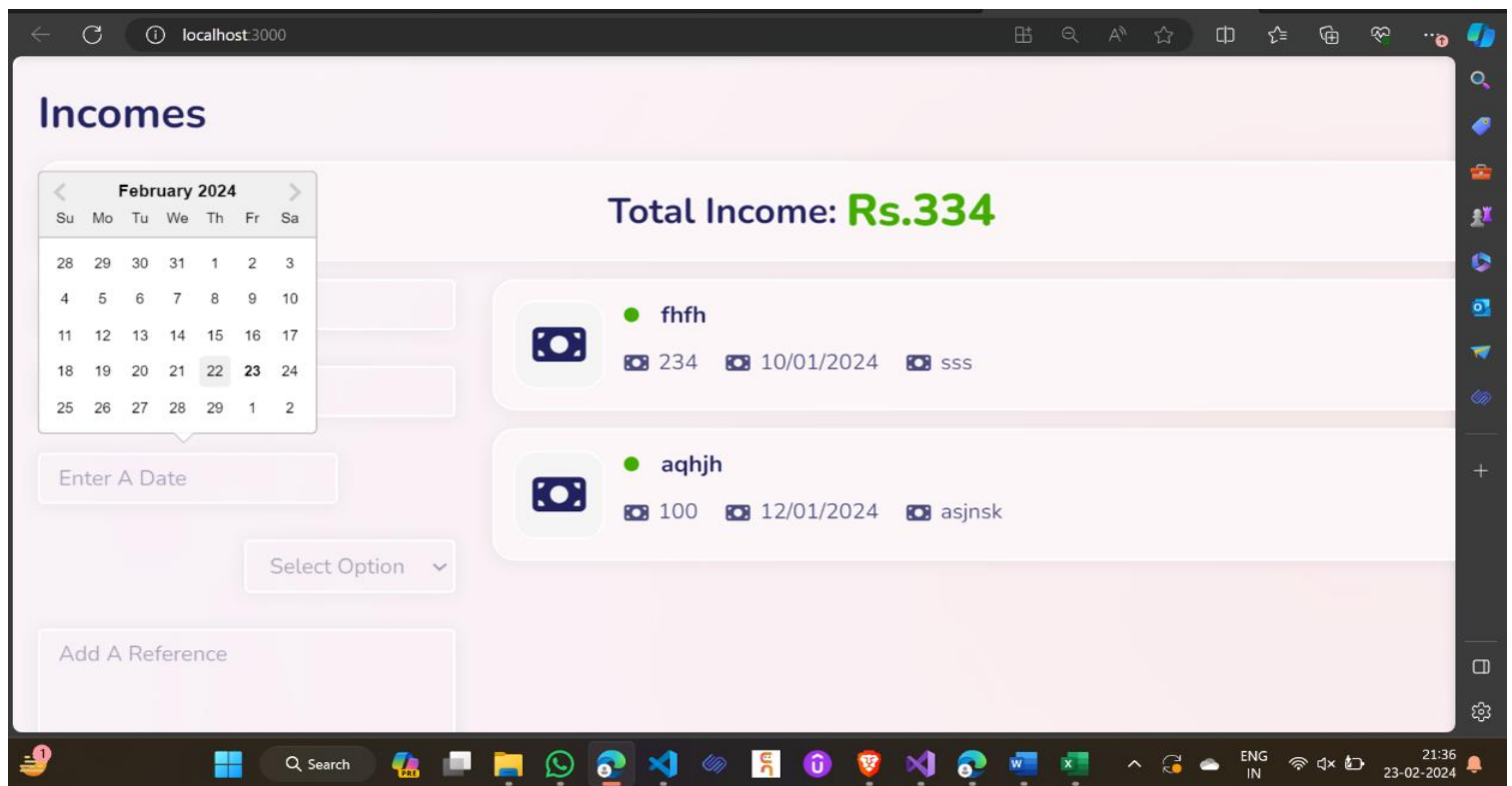
aqhjh
100 12/01/2024 asjnsk

Overall, this financial tracking system displays information about income and expenses, including the amount, date, and category of each transaction. The user can search or filter transactions, and view a summary of the total income for the time period shown. Additional fields may be available for notes or references.

The fields and labels in this system include:

- "Fees": This field may be a category or tag for expenses.
- "Q": This field may be a search or filter function for the system.
- "200": This field is the amount of the most recent expense.
- "Incomes": This field may be a label or title for the income section of the system.
- "Enter A Date": This field is likely where the user can enter a date range for the transactions displayed.
- "Add A Reference": This field may allow the user to add a reference or note to a transaction.
- Total Income: Rs.334": This field is a summary of the total income for the time period shown.
- "10/01/2024" and "12/01/2024": These fields are date ranges for the transactions displayed.

4.3.3 Add Date:



This appears to be a financial tracking system that displays information about income and expenses. The system includes several fields and labels that provide information and allow the user to interact with the system.

The fields and labels in this system include:

"Fees": This field may be a category or tag for expenses.

"Q": This field may be a search or filter function for the system.

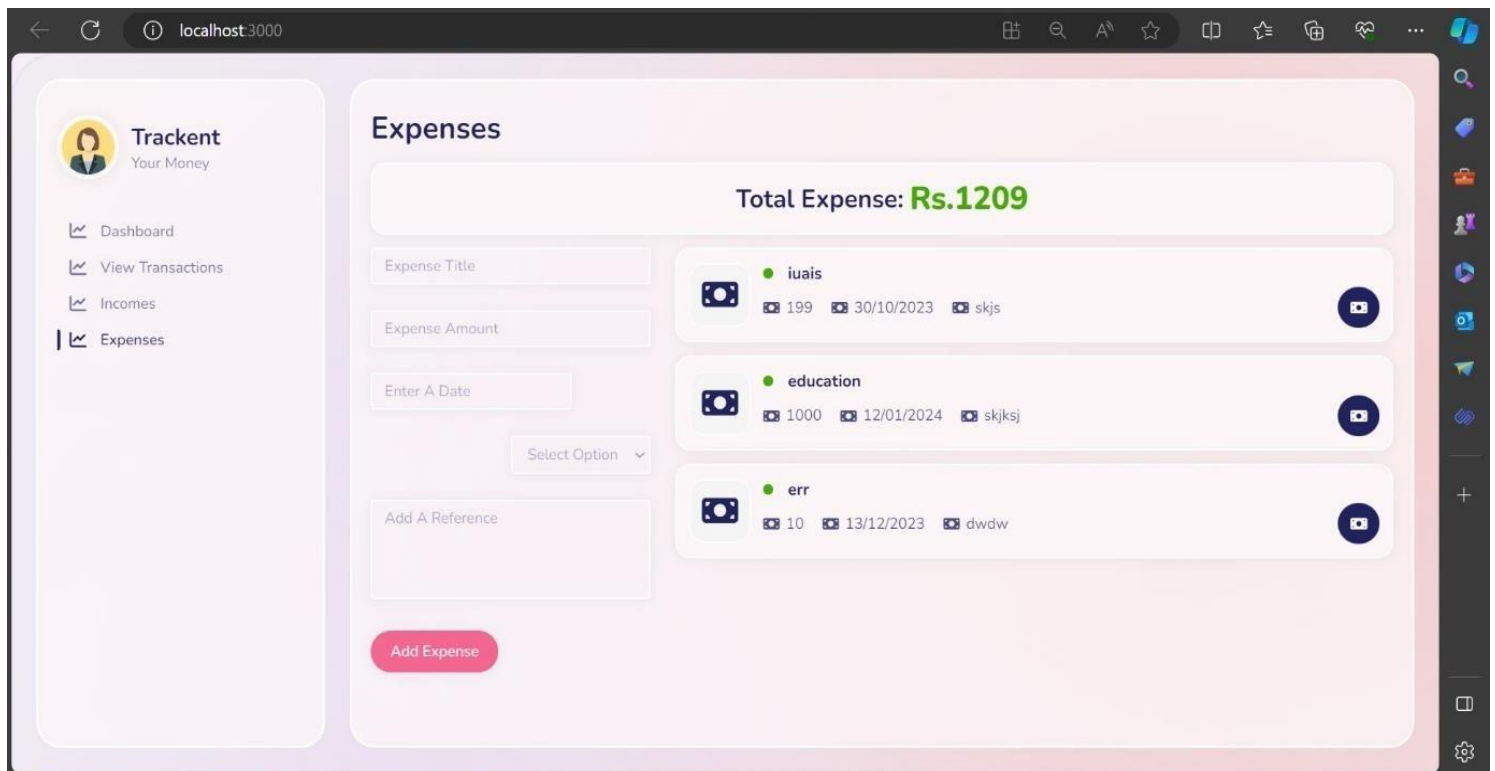
"200": This field is the amount of the most recent expense.

"Incomes": This field may be a label or title for the income section of the system.

"Enter A Date": This field is likely where the user can enter a date range for the transactions displayed.

"Add A Reference": This field may allow the user to add a reference or note to a transaction.

4.4 Expense Module



In an expense tracker system project, the expense module typically refers to the component responsible for managing and tracking expenses incurred by users. Here's how it might work:

User Interface (UI): The expense module usually starts with a user interface where users can input their expenses. This UI might include fields such as date, amount, category (e.g., food, transportation, entertainment), and any additional notes.

Expense Submission: Once users fill out the necessary information, they submit the expense through the UI. The expense module receives this data.

Validation: The expense module validates the submitted expense to ensure that it meets certain criteria. This could include checking for valid date formats, ensuring that the amount is a number, and verifying that required fields are filled out.

Storage: Validated expenses are then stored in a database. The database could be relational (like MySQL or PostgreSQL) or NoSQL (like MongoDB), depending on the project's requirements.

4.4.1 Add Title of Expense:

The screenshot displays the 'Trackent' web application interface. On the left is a sidebar with a user profile icon and the text 'Trackent Your Money'. Below this are navigation links: 'Dashboard', 'View Transactions', 'Incomes', and 'Expenses'. The main content area is titled 'Expenses'. At the top right of this area, a summary bar shows 'Total Expense: Rs.1209'. Below the title, there is a form for adding a new expense. It includes a text input field with 'fess', a numeric input field with '200', and a date input field with the placeholder 'Enter A Date'. A calendar for February 2024 is open, showing dates from 28 to 2. Below the calendar is a pink 'Add Expense' button. To the right of the form, a list of existing expenses is shown. Each entry includes a category icon, a category name, an amount, a date, and a user initials icon. The entries are: 'iuais' (199, 30/10/2023, skjs), 'education' (1000, 12/01/2024, skjksj), and 'err' (10, 13/12/2023, dwdw). The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 21:37 on 23-02-2024.

Overall, this expense module input form allows the user to input and track expenses, including the category, amount, and date of each expense. The user can also view all transactions, search or filter expenses, and view a summary of the total expenses for the time period shown.

The fields and labels in this form include:

- "Expenses": This field may be a label or title for the expense section of the form.
- "fess 200": These fields are the category and amount of the expense being added.
- "Enter A Date": This field is where the user can input the date of the expense.
- "< February 2024": This field is a date range for the expenses displayed.
- "Su Mo Tu We Th": These fields are the abbreviations for the days of the week.
- "Add Expense": This field may be a button or link to add a new expense.
- "199 education 1000 err 10": These fields are the categories and amounts of other expenses.
- "30/10/2023": This field is a date range for the expenses displayed.
- iuais": This field may be a label or title for a specific expense category.
- "Total Expense: Rs.1209": This field is a summary of the total expenses for the time period.

4.4.2 Add Amount and Date of Expense:

The screenshot shows a web application running on localhost:3000. The interface is divided into a left sidebar and a main content area. The sidebar, titled 'Trackent Your Money', contains a user profile icon and a list of navigation links: 'Dashboard', 'View Transactions', 'Incomes', and 'Expenses'. The main content area is titled 'Expenses' and features a form for adding new expenses and a list of existing ones. The form includes input fields for 'fess', '200', and '07/02/2024', a dropdown menu set to 'Education', and a text area with 'XYZ...'. A pink 'Add Expense' button is at the bottom of the form. The list of expenses shows three entries: 'iuais' with amount '199' and date '30/10/2023', 'education' with amount '1000' and date '12/01/2024', and 'err' with amount '10' and date '13/12/2023'. A summary bar at the top of the list states 'Total Expense: Rs.1209'. The Windows taskbar at the bottom shows the time as 21:37 on 23-02-2024.

Category	Amount	Date	Reference
iuais	199	30/10/2023	skjs
education	1000	12/01/2024	skjsj
err	10	13/12/2023	dwdw

Overall, this financial tracking system displays information about income and expenses, including the category, amount, and date of each transaction. The user can view all transactions, search or filter expenses, and view a summary of the total expenses for the time period shown. The system also includes a button or link to add a new expense. Additional fields may be available for notes or references.

"Total Expense: Rs.1209": This field is a summary of the total expenses for the time period shown.

"iuais": This field may be a label or title for a specific expense category.

"199 education 1000 err 10": These fields are the categories and amounts of other expenses.

"30/10/2023": This field is a date range for the expenses displayed.

CHAPTER 5

TESTING

Build up Our project We Use Software Testing Process for executing a program with the intent of finding errors that is uncovering errors in a program makes it a feasible task and also trying to find the errors (whose presence is assumed) in a program. As it is a destructive process. Here we just mentioned that how the testing is related to this software and in which way we have test the software?

In our project we have used 5 types of testing these are listed below

5.1 Unit Testing

Unit testing is a software testing method that focuses on testing individual units or components of a software system, such as functions, methods, or classes. In the context of an expense tracker, unit testing can help ensure that each component of the system is functioning correctly and as intended.

Here are some steps for implementing unit testing in an expense tracker:

1. **Identify the units to be tested:** Identify the individual units or components of the system that need to be tested, such as functions for calculating expenses, data input forms, or financial reports.
2. **Write test cases:** Write test cases for each unit, specifying the expected input, output, and behaviour of the unit.
3. **Implement test code:** Implement the test code using a unit testing framework, such as
4. **Run the tests:** Run the tests and compare the actual results with the expected results.
5. **Analyse and fix issues:** Analyse any issues that arise during testing, and fix the code as necessary.
6. **Repeat testing:** Repeat the testing process for each new release or update of the expense tracker.

By implementing unit testing in an expense tracker, you can ensure that each component of the system is functioning correctly and as intended, and that any issues are identified and fixed early on in the development process. This can help improve the overall quality and reliability of the system, and reduce the risk of bugs and errors.

5.2 Module Testing

Module testing, also known as integration testing, is a software testing method that focuses on testing the interactions between different modules or components of a software system. In the context of an expense tracker, module testing can help ensure that the different modules of the system are working together correctly and as intended.

Here are some steps for implementing module testing in an expense tracker:

1. Identify the modules to be tested: Identify the different modules or components of the system that need to be tested, such as the user interface, database, and financial reporting modules.
2. Define the interfaces between modules: Define the interfaces between the modules, specifying the expected input, output, and behaviour of each module.
3. Write test cases: Write test cases for each module, specifying the expected input, output, and behaviour of the module when interacting with other modules.
4. Implement test code: Implement the test code using a module testing framework.
5. Run the tests: Run the tests and compare the actual results with the expected results.
6. Analyse and fix issues: Analyse any issues that arise during testing, and fix the code as necessary.
7. Repeat testing: Repeat the testing process for each new release or update of the expense tracker.

By implementing module testing in an expense tracker, you can ensure that the different modules of the system are working together correctly and as intended, and that any issues are identified and fixed early on in the development process. This can help improve the overall quality and reliability of the system, and reduce the risk of bugs and errors.

It's important to note that module testing should be performed after unit testing, as it helps to identify issues that may arise when modules are integrated together.

5.3 Sub-system Testing

Subsystem testing, also known as system integration testing, is a software testing method that focuses on testing the interactions between different subsystems or subsets of a software system. In the context of an expense tracker, subsystem testing can help ensure that the different subsystems of the system are working together correctly and as intended.

By implementing subsystem testing in an expense tracker, you can ensure that the different subsystems of the system are working together correctly and as intended, and that any issues are identified and fixed early on in the development process. This can help improve the overall quality and reliability of the system, and reduce the risk of bugs and errors.

It's important to note that subsystem testing should be performed after module testing, as it helps to identify issues that may arise when subsystems are integrated together.

Subsystem testing is a critical step in the software development process, as it helps to ensure that the different parts of the system are working together seamlessly, and that the system as a whole is functioning as intended. By investing time and resources in subsystem testing, you can improve the overall quality and reliability of your expense tracker, and provide a better user experience for your customers.

5.4 System Testing

By implementing system testing in an expense tracker, you can ensure that the system is functioning correctly and as intended, and that it meets the requirements and expectations of its users. This can help improve the overall quality and reliability of the system, and reduce the risk of bugs and errors.

It's important to note that system testing should be performed after subsystem testing, as it helps to identify issues that may arise when the system is used as a whole. System testing should also take into account the user experience, and should test the system in a realistic environment, with real data and real-world use cases.

By investing time and resources in system testing, you can improve the overall quality and reliability of your expense tracker, and provide a better user experience for your customers.

It's also important to note that system testing should be performed by an independent team, separate from the development team, to ensure that the testing is unbiased and thorough. This can help to ensure that the system meets the needs and expectations of its users, and that it is free from bugs and errors.

In conclusion, system testing is a critical step in the software development process, as it helps to ensure that the system is functioning correctly and as intended, and that it meets the requirements and expectations of its users. By investing time and resources in system testing, you can improve the overall quality and reliability of your expense tracker, and provide a better user experience for your customers.

5.5 Acceptance Testing

By implementing acceptance testing in an expense tracker, you can ensure that the system is functioning as intended and that it meets the needs of its users. This can help improve the overall quality and reliability of the system, and reduce the risk of bugs and errors.

It's important to note that acceptance testing should be performed by the users themselves, or by a team that represents the users, to ensure that the testing is unbiased and representative of how the system will be used in production. This can help to ensure that the system meets the needs and expectations of its users, and that it is free from bugs and errors.

In conclusion, acceptance testing is a critical step in the software development process, as it helps to ensure that the system is functioning as intended and that it meets the requirements and expectations of its users. By investing time and resources in acceptance testing, you can improve the overall quality and reliability of your expense tracker, and provide a better user experience for your customers.

It's also important to note that acceptance testing should be performed in the final stages of the software development process, after all other testing has been completed, to ensure that the system is ready for production use. By involving the users in the acceptance testing process, you can ensure that the system meets their needs and expectations, and that it is ready for deployment.

CONCLUSION

An expense tracker system is a financial tool that allows users to input, track, and monitor their expenses over a period of time. The system typically includes features such as a form for inputting expenses, a chart or graph displaying expenses, and options for searching or filtering transactions. The system may also include a summary of the total expenses for a specific time period, as well as a minimum amount required for a transaction.

The expense tracker system can help users manage their finances by providing a clear and organized view of their expenses. By inputting and tracking expenses, users can identify areas where they may be overspending and make adjustments to their budget. The system can also help users stay accountable for their spending and make informed decisions about their finances.

Overall, an expense tracker system is a valuable tool for anyone looking to take control of their finances and make informed decisions about their spending.

REFERENCES

- [1] Online Income and Expense Tracker S. Chandini¹, T. Poojitha², D. Ranjith³, V.J. Mohammed Akram⁴, M.S. Vani⁵, V.
- [2] Expense Tracker : A Smart Approach to Track Everyday Expense Hrithik Gupta school of
- [3] Online Income and Expense Tracker S. Chandini¹, T. Poojitha², D. Ranjith³, V.J. Mohammed Akram⁴, M.S. Vani⁵, V.
- [4] Expense Tracker ATIYA KAZI¹ , PRAPHULLA S. KHERADE² , RAJ S. VILANKAR³ , PARAG M. SAWANT⁴ 1 Professor.

BIBLIOGRAPHY

- [1] L.T.HONG, "Android Mobile Application-Expenses with Geo-Location Tracking, "University Tunku Abdul Rahman, UTAR,2015,
- [2] N. Mehra, "Android Application Expense Manager," 2014.
- [3] A. Bamne, "expense tracker(class Diagram(UML)," 2014.

