

PROJECT REPORT

FINANCELY

Table of CONTENTS

01

Introduction

Project Overview
Purpose and Motivation
Objectives

02

Key Features

User Registration and Authentication
Transaction Management
Pagination for Transaction Records
Budget Tracking
Visual Statistics and Charts

03

Technology Stack

Frontend
Backend
Database

04

Data Structures Used

Doubly Linked List
Priority Queue
Hash Maps
KMP Algorithm

05

Functional Workflow

User Registration and Login
Transaction Management
Setting and Monitoring Budgets
Data Visualization and Statistics



ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Dip Sankar Banerjee, whose guidance, expertise, and encouragement were invaluable throughout the development of this project. His insightful feedback and thoughtful discussions helped me refine my approach and gain a deeper understanding of key concepts, for which I am incredibly grateful.

I also extend my thanks to Ms. Neha Sharma, for their patience, assistance, and constructive feedback on both technical and conceptual aspects of the project. Their support and prompt responses to questions made navigating challenges much more manageable and greatly contributed to the project's success. Thank you both for your dedication and support throughout this journey.

INTRODUCTION

Project Overview

1

Financely is a personal expense tracking application designed to help individuals manage their finances by recording and categorizing expenses. By offering interactive features such as dynamic charts, budget setting, and pagination for large datasets, Financely aims to simplify financial monitoring. This application allows users to have a holistic view of their spending patterns and budget usage, enabling them to take control of their finances with minimal effort.

2

Purpose and Motivation

In today's world, managing personal finances is increasingly essential for individuals seeking financial stability. Traditional methods of tracking expenses manually or through spreadsheets are tedious and error-prone. Financely was developed to provide a modern, accessible tool that empowers users to maintain financial discipline by offering them insights into their spending habits. It is tailored for individuals who want a simple, efficient, and visually appealing solution to track expenses and manage budgets.

3

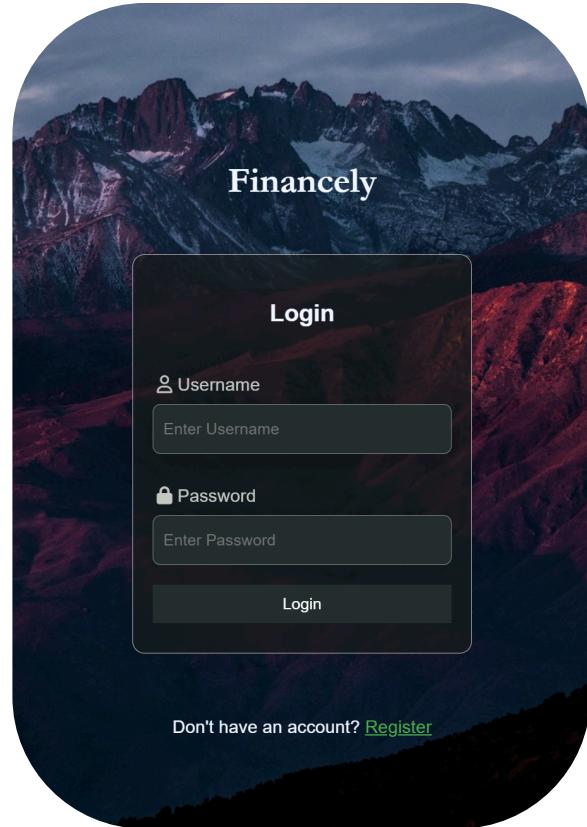
Objectives

- To create a user-friendly and intuitive platform for expense tracking.
- To provide detailed expense categorization and budget management features.
- To visualize spending trends using interactive graphs and charts.
- To manage large transaction datasets efficiently using pagination.
- To ensure data security and scalability for a growing user base.

KEY FEATURES

USER REGISTRATION AND AUTHENTICATION

- Secure user authentication is a critical aspect of the application, ensuring that each user's data is private and accessible only by them.
- A registration system allows users to create accounts by providing a username and password, while the login system uses encrypted passwords to authenticate users securely.

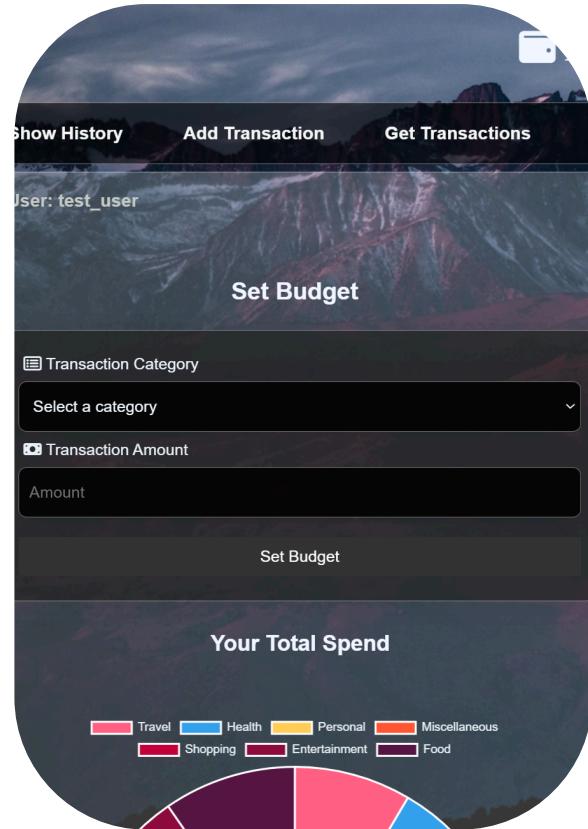
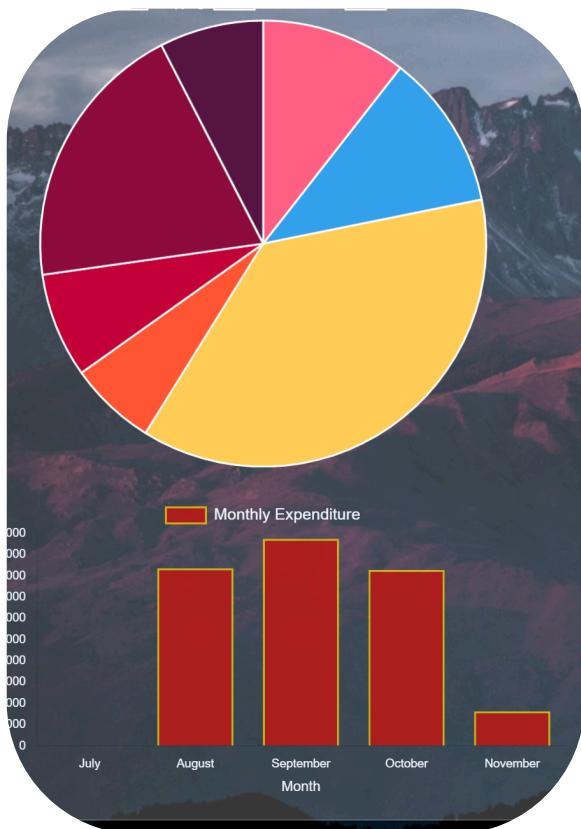


TRANSACTION MANAGEMENT

- Users can easily add, view, edit, and delete transactions. Each transaction records details like date, amount, category, and description.
- Transactions are categorized by labels (e.g., "Food," "Transportation," "Entertainment"), which helps users analyze spending in different areas
- Large datasets are paginated, ensuring smooth navigation and better performance.

BUDGET TRACKING

- Users can set budgets on specific categories (e.g., 200 INR for Food).
- The system then tracks the user's spending in each category and compares it to the set budget, notifying the user when they are close to exceeding their budget.
- This feature provides an efficient way for users to stay within their financial limits.



VISUAL STATISTICS AND CHARTS

- Monthly and category-based expenses are visualized through bar and pie charts created with Chart.js, giving users a clear view of their spending trends.
- The charts are interactive and responsive, updating dynamically as users add, edit, or delete transactions.

Technology Stack

FRONTEND

- HTML5 & CSS3: Provide the structure and styling of the web application, ensuring a clean and responsive interface.
- JavaScript: Handles interactivity, such as pagination, chart updates, and form submissions.
- Chart.js: A JavaScript library used for creating interactive bar and pie charts to display expense data visually, making it easier for users to understand their spending patterns at a glance.
- Font Awesome: Used for icons, improving the visual appeal of UI elements like buttons, forms, and navigation.

BACKEND

- Node.js with Express.js: Node.js is used for the server-side logic, while Express.js provides the necessary framework for handling routing and creating RESTful APIs.
- C++: Responsible for handling core transaction functionalities, data processing, and storage. C++ was chosen to perform intensive tasks due to its efficiency.

DATABASE

- Currently, Financely uses a file-based data storage system in C++ to manage and store transaction data.

Data Structures Used

DOUBLY LINKED LIST

- A doubly linked list is used to store the transactions and used to further add new transactions as well as for the retrieval of transactions as per the user filter

PRIORITY QUEUE

- A priority queue is used to keep track of overspending in a sorted order so as to warn the user as per the amount overspend

HASH MAPS

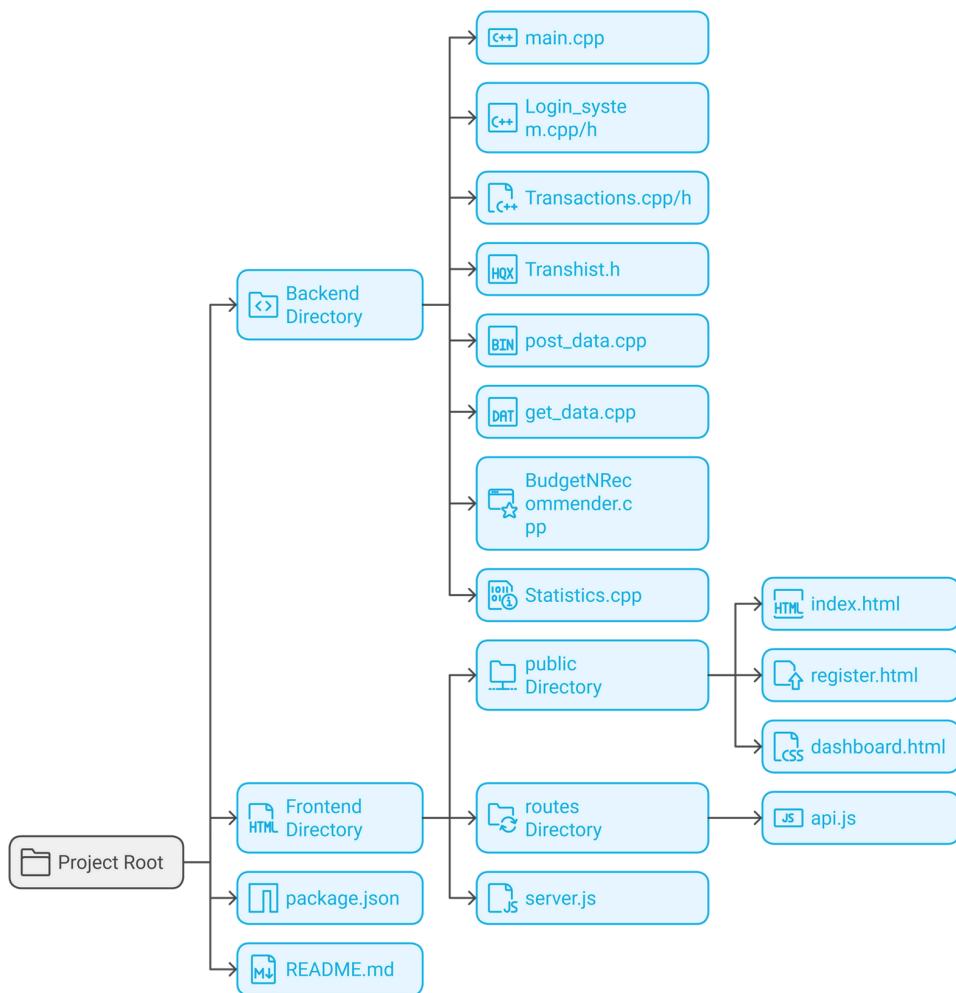
- Unordered maps are used to store data categorically so as for efficient retrieval in the case of categorical data retrieval

KMP ALGORITHM

- KMP Algorithm is used for string matching as to find a particular key in the whole transaction string stored

FUNCTIONAL WORKFLOW

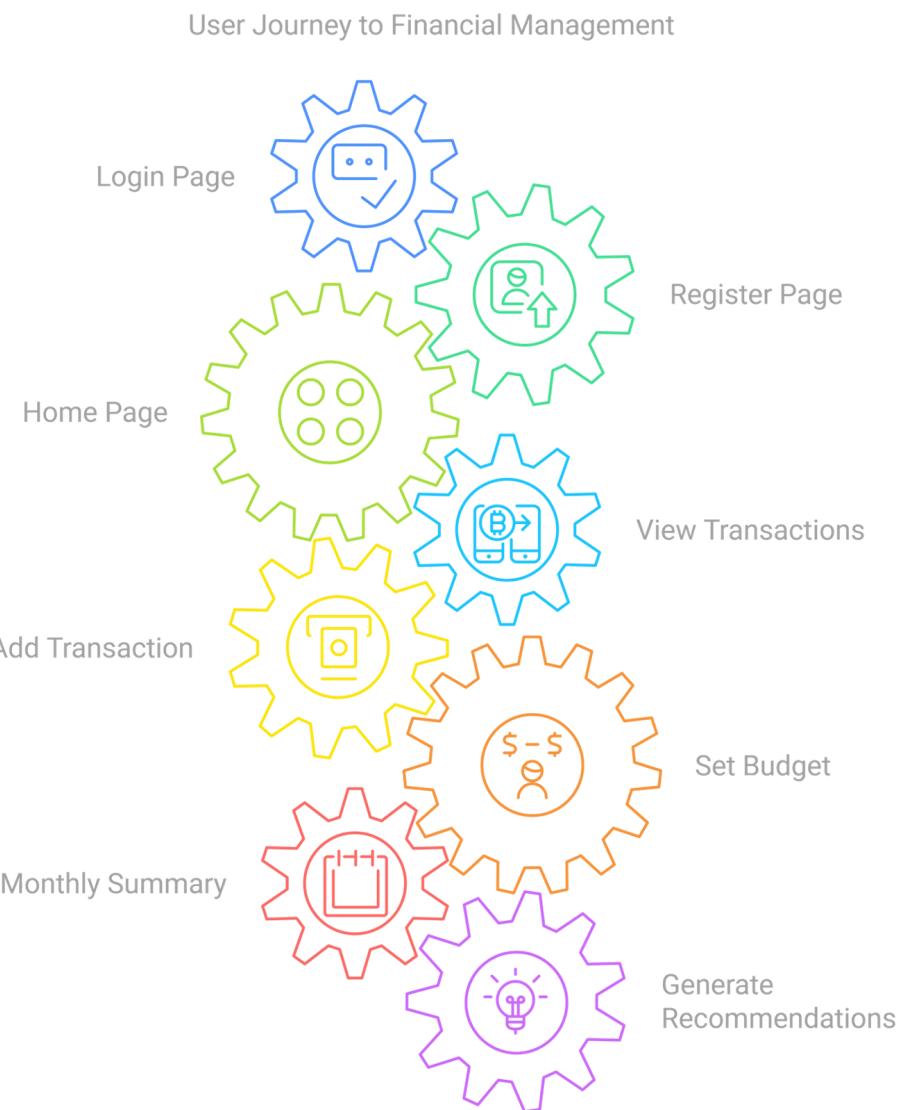
PROJECT STRUCTURE



Backend Directory: Contains all the backend cpp files managing the main functioning of the project such as transaction adding and storing

Frontend Directory: The public folder contains the frontend html,css and javascript that is served to the users. `Server.js` is the main expressjs server acting as a bridge between the frontend and the backend connecting both seamlessly.

USER FLOW



Login/Register Page: As the user enters our website, he/she encounters the login page which has the link for the register page for the first time users. The user logs in or registers securely managed by the Login_system.cpp file

Main Dashboard: After the login, the user encounters the main dashboard which features few last transactions, spending pie chart, monthly spend bar graph which is managed by the get_data.cpp file



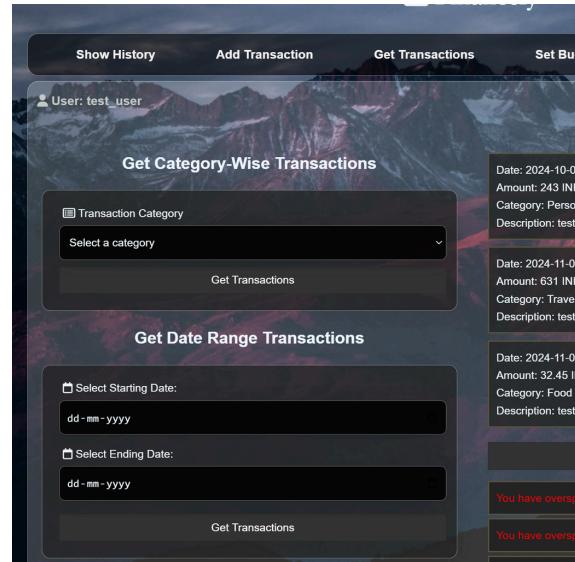
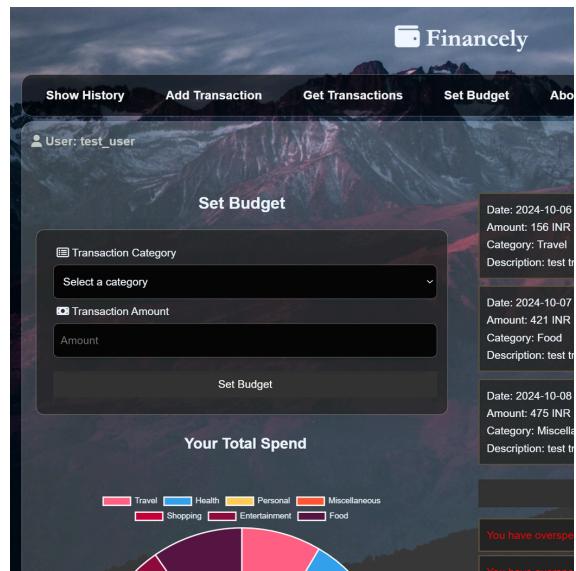
The screenshot shows the 'Add Transaction' interface. At the top, it says 'User: test_user'. Below that is a large input field for 'Add Transaction'. Inside, there's a date picker, a text input for 'Transaction Amount', a dropdown for 'Transaction Category', a dropdown for 'Short Description', and a note area with the message 'You have over spent'.

Add Transactions: This tab on click displays a form which allows the user to add a new transaction, the transactions details is then added to the doubly linked list for further storing. This is managed by post_data.cpp file

Show History: This tab on click displays all the user transactions till date. Since the number of transactions can be huge, the transactions are paginated. This is maintained by get_data.cpp file

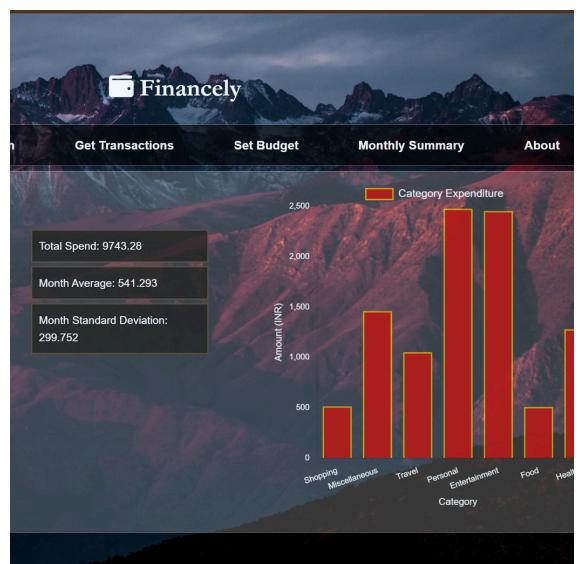
Transactions			
Date: 2024-10-01 Amount: 478 INR Category: Health Description: test transaction	Date: 2024-10-01 Amount: 500 INR Category: Shopping Description: test transaction	Date: 2024-10-02 Amount: 824 INR Category: Entertainment Description: test transaction	Date: 2024-10-02 Amount: 824 INR Category: Personal Description: test transaction
Date: 2024-10-02 Amount: 574.32 INR Category: Entertainment Description: test transaction	Date: 2024-10-02 Amount: 964.7 INR Category: Personal Description: test transaction	Date: 2024-10-03 Amount: 761 INR Category: Entertainment Description: test transaction	Date: 2024-10-03 Amount: 761 INR Category: Shopping Description: test transaction
Date: 2024-10-04 Amount: 655 INR Category: Travel Description: test transaction	Date: 2024-10-04 Amount: 655 INR Category: Personal Description: test transaction	Date: 2024-10-05 Amount: 478 INR Category: Health Description: test transaction	Date: 2024-10-05 Amount: 533.14 INR Category: Health Description: test transaction
Date: 2024-10-06 Amount: 156 INR	Date: 2024-10-06 Amount: 100 INR	Date: 2024-10-07 Amount: 421 INR	Date: 2024-10-07 Amount: 142.54 INR

Set Budget: This tab allows the user to set new budget which is stored and further used in the priority queue to generate user recommendations. This is managed by the BudgetNRecommender.cpp file



Get Transactions This tab helps the user to get the transactions as per his/her needs. The user can get the data filtered by category or in a particular range of dates as per the user needs. This is managed by the get_data.cpp file

Monthly Summary: This tab helps the user to get monthly summary of his/her spending. The user can see total spend, average spend and a standard deviation of his/her spend for the month. It also features a bar graph for category wise spend. This is managed by the Statistics.cpp file





THANK YOU



[Github](#)

TEAM MEMBERS

LUV VALECHA B23CS1093

KRISH TECKCHANDANI B23CS1092

ANSHIT AGARWAL B23CS1087

