**X-Track: A Full Stack Expenses Tracker**

**SYNOPSIS/Report**

OF MINI PROJECT

**BACHELOR OF COMPUTER APPLICATIONS**

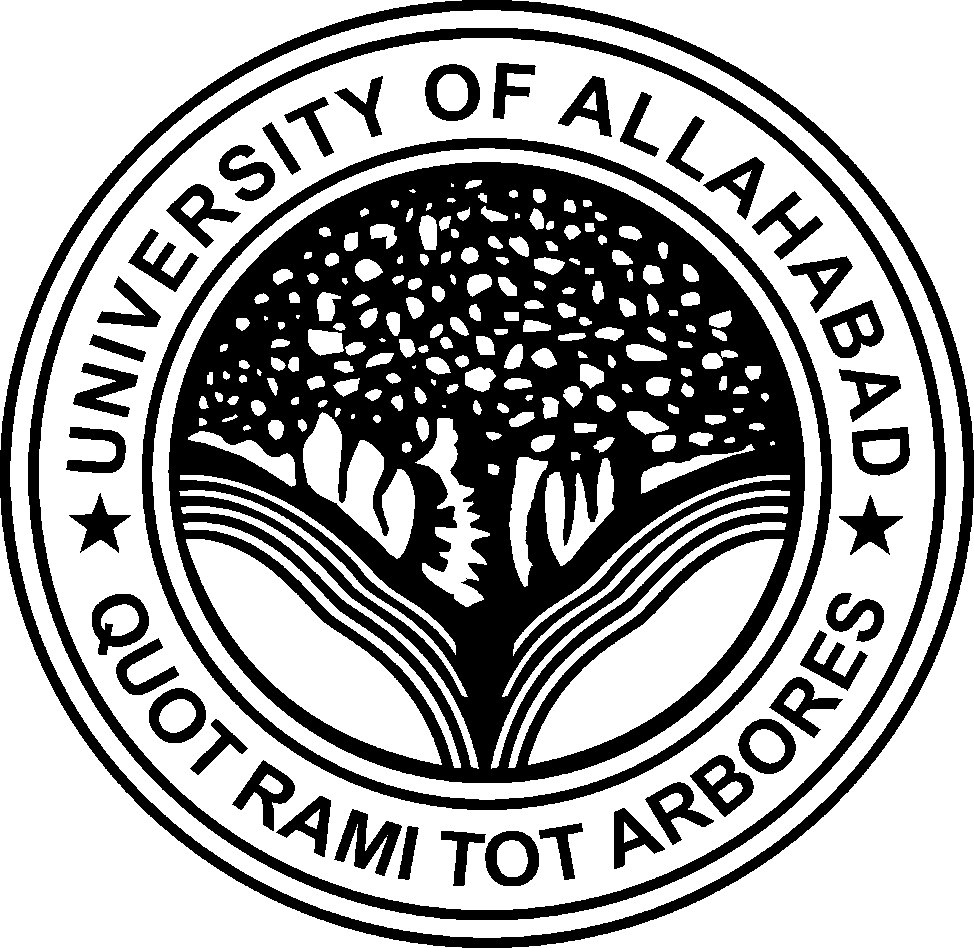
SUBMITTED BY

SHOBHIT PANDEY

Batch Year –2023-26

Enrollment No. -U2346006

**PROJECT GUIDE –BRIJESH SIR**



**Centre of Computer Education & Training Institute of Professional Studies**

**University of Allahabad, Prayagraj Uttar Pradesh-211002**

**INTRODUCTION**

In today's dynamic economic landscape, effective personal financial management has emerged as a critical life skill. For a significant segment of the Indian population, particularly the middle-class and dedicated competitive examination candidates, daily financial decisions directly affect their long-term stability and aim.

Many individuals still dependent on manual methods, mental accounting, or spreadsheets, all of which lead to inaccuracies, lack real-time insights, and fail to provide a wide picture of spending patterns. This factor often leads to unseen financial pressures, missed opportunities for savings, and a general disconnect from one's true financial standing. The "Expense Tracker" project is helping to bridge this gap, offering a modern, innovative, and effective solution.

**Category of the project**: OOPs-based

**PROBLEM DEFINITION**

The main problem addressed by this project starts from the problems that middle-class individuals and competitive exam aspirants face in maintaining a record of their everyday financial transactions. Without a clear view of where their money goes, it becomes challenging to:

* Identify unnecessary spending areas.
* Follow the declared budgets.
* Collect savings for future goals (e.g., higher education, home purchase, emergency fund).
* Understand the impact of small, seemingly unused expenditures.

Existing accounting software often becomes too complex or expensive for this work, while basic mobile notes or calculator apps lack the capabilities required for genuine financial details.

**Motivation –**

For many people, keeping track of expenses can feel overwhelming and confusing, leading to stress and missed chances to save. That is the main reason for this project. We want to build a simple, easy-to-use tool that helps people see exactly where their money is going. Our goal is to make managing your money feel less like a chaos and more like a way to take control of your finances, helping you feel more confident and secure.

**OBJECTIVE(S)**

* The aim of the "Expense Tracker" is to empower its target users with enhanced financial visibility and control.
* Design and implement an interesting and responsive mobile interface using KivyMD.
* Enable users to add, edit, and delete expenses with details such as amount, date, time, category, and notes, ensuring data integrity in MySQL.
* Provide functionality for predefined and custom expense categories, allowing users to efficiently classify their spending.
* Implement robust data storage using MySQL and ensure basic data security.

**Target Audience Profile**

* Middle-Class Families
* Competitive Examination Candidates

**Anticipated Benefits**

The successful deployment of the "Expense Tracker" is expected to deliver benefits to its users:

* Enhanced Financial Awareness
* Improved Budget Adherence
* Increased Savings Potential:
* Reduced Financial Stress:
* Simplicity and Accessibility

**REQUIREMENT ANALYSIS**

**Software Analysis**

* **Frontend:** Kivymd module of Python

(KivyMD, chosen for its Pythonic nature, rich UI components, and cross-platform capabilities. The UI will be broken down into reusable components. Dedicated screens for Dashboard, Add Expense, Categories, Budgets, Reports, and Settings)

* **Backend:** Django module of Python

(Django will receive HTTP requests (GET, POST, PUT, DELETE) from the KivyMD frontend. Use of a Python MySQL connector library to interact with the MySQL database. Robust error handling for API requests)

* **Database:** MySQL (an open-source RDBMS, A MySQL database responsible for storing all application data)
* **Version Control:** Git will be used for source code management, hosted on platforms like GitHub or GitLab, to manage changes, track history, and enable collaborative development.

**Study Design**

The project will develop in an iterative and collaborative approach to software development that breaks projects into small, manageable increments called sprints, allowing for continuous feedback and adaptation to change.

Development will be broken down into a series of short sprints, with each sprint focusing on a specific set of features.

This iterative process will allow for continuous feedback and adaptation.

It prioritizes customer satisfaction through frequent delivery of working software, fostering a flexible, self-organizing team environment with strong communication and collaboration.

**Type of Problem (clustering/ classification/ regression/ Translation/ Association rule mapping) and domain (Vision, Text, Big Data, Data science, others, hybrid)**

The primary problem type is **Data Management and Visualization**. The core domain is **RDBMS / Data Science**. While the initial version will not include complex machine learning, the foundation is laid for future enhancements, such as using regression or classification algorithms to predict future spending patterns or identify potential savings.

**Name of Algorithm(s) developed or used**

For the core functionality, the primary algorithms are standard **CRUD (Create, Read, Update, Delete)** operations on the database. Data aggregation algorithms will be used for calculating sums, averages, and displaying totals for reports. Sorting and filtering algorithms will be used to organize transaction lists.

**Testing strategy used**

A multi-layered testing strategy will be employed:

* **Unit Testing:** To ensure individual functions and components work as expected.
* **Integration Testing:** To verify that different modules (e.g., frontend to backend, backend to database) work together seamlessly.
* **User Acceptance Testing (UAT):** The final application will be tested by a small group of users to validate that it meets the stated objectives and is intuitive to use.

**[Start of Document Content]**

**Project Proposal & Preliminary Design Document**

**Project Title:** Expense Tracker

Submitted By: [Shobhit Pandey]

Submission Date:, 2025

**Table of Contents**

1. Introduction

1.1. Problem Statement

1.2. Existing Solutions & Their Limitations

1. Project Goals and Objectives

2.1. Overall Goal

2.2. Specific Objectives

1. Target Audience and Expected Benefits

3.1. Target Audience Profile

3.2. Anticipated Benefits

1. Scope of the Project

4.1. In-Scope Features

4.2. Out-of-Scope Features (Future Enhancements)

1. Detailed Feature Breakdown

5.1. Expense Logging Module

5.2. Category Management Module

5.3. Budgeting Module

5.4. Reporting & Visualization Module

5.5. Search & Filter Functionality

5.6. Data Persistence and Security Considerations

1. Technical Architecture and Design

6.1. System Overview

6.2. Frontend Architecture (KivyMD)

6.3. Backend Architecture (Flask)

6.4. Database Design (MySQL)

6.5. Data Flow Diagram (Conceptual)

1. Methodology and Development Process

7.1. Development Approach

7.2. Project Phases

1. System Requirements

8.1. Software Requirements

8.2. Hardware Requirements

1. Conclusion
2. References (Optional, if you cite any statistics or concepts)

**1. Introduction**

1.1. Problem Statement

1.2. Existing Solutions & Their Limitations

While various expense tracking applications exist in the market, many are either overly complex, laden with premium features irrelevant to daily tracking, or lack a user interface tailored to the simplicity and speed required by individuals managing moderate incomes and busy schedules. Some solutions are designed for broad financial planning rather than granular daily expenditure tracking. Furthermore, data privacy concerns and the preference for locally controlled data (or simple cloud sync) are significant factors for the target audience. The proposed "Expense Tracker" aims to offer a focused, efficient, and user-centric alternative.

**2. Project Goals and Objectives**

2.1. Overall Goal

To develop a robust, intuitive, and cross-platform mobile application that simplifies the process of daily expense logging, categorization, and analysis, thereby enabling middle-class individuals and competitive exam candidates to achieve greater financial awareness, discipline, and effective budget management.

2.2. Specific Objectives

To realize the overall goal, the project will strive to meet the following SMART (Specific, Measurable, Achievable, Relevant, Time-bound) objectives:

* **O1: Develop a User-Friendly Interface:** by [Target Date 1, e.g., End of Month 2] that allows users to log an expense in under 10 seconds.
* **O2: Implement Core Expense Management:** by [Target Date 2, e.g., Mid-Month 3].
* **O3: Enable Comprehensive Categorization:** by [Target Date 3, e.g., End of Month 3].
* **O4: Integrate Basic Budgeting Features:** Allow users to set monthly budgets for specific categories and visualize their spending progress against these budgets by [Target Date 4, e.g., Mid-Month 4].
* **O5: Generate Actionable Reports:** Develop capabilities to generate summary reports (daily, weekly, monthly) and visual charts (e.g., pie, bar) of spending patterns by [Target Date 5, e.g., End of Month 4].
* **O6: Ensure Data Persistence & Basic Security:** (e.g., preventing unauthorized access to personal data on the device) by [Target Date 6, e.g., Mid-Month 5].

**3. Target Audience and Expected Benefits**

The "Expense Tracker" is meticulously designed with a clear understanding of its primary users and their unique needs.

**3.1.**

**4. Scope of the Project**

This section delineates the boundaries of the "Expense Tracker" project for its initial academic submission, clearly stating what will be included and what is considered for future development.

4.1. In-Scope Features

The Minimum Viable Product (MVP) for this academic project will focus on the following core functionalities:

* **User Management (Basic):** Local user profiles on the device; no cloud synchronization in the initial phase.
* **Expense Entry:** Manual input of transaction details including amount, date, time, pre-defined/custom category, and a brief description.
* **Categorization:** Creation, editing, and deletion of custom expense categories.
* **Basic Budgeting:** Setting monthly budgets per category and displaying progress against these budgets.
* **Reporting:** Generation of textual summary reports (daily, weekly, monthly) and basic pie/bar charts showing spending distribution by category for a selected period.
* **Data Storage:** Local data persistence using a MySQL database (accessed via Flask backend).
* **Search & Filter:** Basic filtering of expenses by date range and category.

4.2. Out-of-Scope Features (Future Enhancements)

While not part of the initial academic scope, the following features are envisioned for future iterations of the "Expense Tracker":

* **Cloud Synchronization:** Securely sync data across multiple devices and provide cloud backup.
* **Income Tracking:** Ability to record and manage various sources of income alongside expenses.
* **Debt Management:** Features for tracking loans, EMIs, and debt repayment schedules.
* **Bill Reminders:** Automated reminders for upcoming bill payments.
* **Advanced Analytics:** Predictive insights based on spending patterns, anomaly detection.
* **Multi-Currency Support:** For users dealing with different currencies.
* **Receipt Scanning:** OCR-based functionality to automatically extract expense details from receipts.
* **Integration with Financial Institutions:** Secure API integration with bank accounts or credit cards (requires significant security and regulatory considerations).
* **Web Interface:** A complementary web application for desktop access.

**5. Detailed Feature Breakdown**

This section provides a more granular look into the functionalities proposed for the "Expense Tracker."

**5.1. Expense Logging Module**

* **Quick Add Interface:** A prominent button or field on the main dashboard for rapid entry of amount and category.
* **Detailed Entry Form:** A dedicated screen for adding comprehensive details including:
  + **Amount:** Numeric input.
  + **Date & Time:** Date picker and time picker for accurate logging.
  + **Category:** Dropdown or search box linked to defined categories.
  + **Description/Notes:** Free-text field for additional context.
  + **Payment Method (Optional for MVP):** e.g., Cash, Debit Card, Credit Card (can be added if time permits).
* **Edit/Delete Functionality:** Users can modify or remove existing expense entries.

**5.2. Category Management Module**

* **Predefined Categories:** A standard set of common expense categories (e.g., Food, Transport, Rent, Education, Entertainment, Health).
* **Custom Category Creation:** Users can define their own categories to tailor the application to their specific needs.
* **Category Management Interface:** Screens to view, edit, and delete user-defined categories.
* **Sub-Categorization (Future):** Ability to nest categories (e.g., "Food" -> "Groceries", "Dining Out").

**5.3. Budgeting Module**

* **Budget Setting:** Users can specify a maximum spending limit for each category (e.g., "₹5000 for Food" per month).
* **Budget Period Selection:** Primarily monthly budgets, with potential for weekly.
* **Progress Visualization:** A progress bar or percentage display showing current spending against the budget for each category.
* **Alerts/Warnings:** Visual cues or simple notifications when a user approaches or exceeds their budget for a specific category.

**5.4. Reporting & Visualization Module**

* **Summary Dashboard:** A main screen displaying key metrics like total spending for the current day/week/month, top spending categories.
* **Tabular Reports:** Generation of lists of expenses filtered by date range, category, or amount.
* **Graphical Reports:**
  + **Pie Charts:** To visualize spending distribution across different categories.
  + **Bar Charts:** To compare spending in different categories or across different months/weeks.
  + **Line Graphs (Future):** To show spending trends over time.
* **Export Options (Future):** The ability to export reports in common formats like CSV or PDF for external analysis or record-keeping.

**5.5. Search & Filter Functionality**

* **Date Range Filtering:** View expenses for a specific day, week, month, year, or custom date range.
* **Category Filtering:** Filter expenses to show only those belonging to a specific category.
* **Keyword Search:** Search expense descriptions for specific terms.
* **Amount-Based Filtering:** Filter expenses based on a minimum or maximum amount.

**5.6. Data Persistence and Security Considerations**

* **Local Storage:** All expense data will be stored locally on the user's device in a MySQL database.
* **Data Integrity:** Measures to ensure data accuracy and consistency during read/write operations.
* **Basic Security:** Focus on securing the database files on the device and ensuring proper API authentication between frontend and backend. Encryption of sensitive data within the database will be considered for future enhancements.

**6. Technical Architecture and Design**

The "Expense Tracker" will adopt a client-server architecture, with the KivyMD application acting as the client and a Flask application serving as the backend API. MySQL will serve as the persistent data store.

6.1. System Overview

The system will comprise three primary layers:

* **Presentation Layer (Frontend):** The KivyMD mobile application, responsible for user interaction, data input, and displaying information.
* **Application Logic/API Layer (Backend):** A Flask web service that handles business logic, processes requests from the KivyMD app, and interacts with the database.
* **Data Layer (Database):** (users, expenses, categories, budgets).

(Conceptual Data Flow Diagram - You can draw this if presenting, or describe it here):

User interacts with KivyMD App -> KivyMD App sends API requests (HTTP/HTTPS) to Flask Backend -> Flask Backend processes requests, interacts with MySQL Database -> MySQL Database stores/retrieves data -> Flask Backend sends API responses to KivyMD App -> KivyMD App displays data to User.

**6.2. Frontend Architecture (KivyMD)**

* **Framework:**
* **Component-Based Design:** (e.g., custom widgets for expense cards, budget progress bars).
* **Screens:**

**6.3. Backend Architecture (Flask)**

* **Framework:** Flask, a lightweight Python micro-framework, ideal for building RESTful APIs.
* **RESTful API Endpoints:** Defined endpoints for each core operation (e.g., /expenses for getting/adding expenses, /categories for managing categories).

6.4. Database Design (MySQL)

The relational database will store the application's data using a structured schema.

* **users Table:**
  + user\_id (Primary Key, INT)
  + username (VARCHAR)
  + password\_hash (VARCHAR) - *Password hashing is crucial for security.*
  + created\_at (TIMESTAMP)
* **categories Table:**
  + category\_id (Primary Key, INT)
  + user\_id (Foreign Key to users.user\_id, INT) - *To allow user-specific categories.*
  + name (VARCHAR)
  + description (TEXT, Optional)
* **expenses Table:**
  + expense\_id (Primary Key, INT)
  + user\_id (Foreign Key to users.user\_id, INT)
  + amount (DECIMAL)
  + date (DATE)
  + time (TIME, Optional)
  + category\_id (Foreign Key to categories.category\_id, INT)
  + description (TEXT)
  + created\_at (TIMESTAMP)
* **budgets Table:**
  + budget\_id (Primary Key, INT)
  + user\_id (Foreign Key to users.user\_id, INT)
  + category\_id (Foreign Key to categories.category\_id, INT)
  + amount\_ budgeted (DECIMAL)
  + start\_date (DATE)
  + end\_date (DATE)
  + created\_at (TIMESTAMP)

6.5. Data Flow Diagram (Conceptual)

(Imagine a simple block diagram here)

* **User Interface (KivyMD):** Input forms, display panels, charts.
* **API Client (KivyMD):** Modules within KivyMD app responsible for making HTTP requests.
* **RESTful API (Flask):** Endpoints (/users, /expenses, /categories, /budgets) handling requests.
* **Business Logic (Flask):** Data validation, calculations, authentication logic.
* **Database Connector (Flask):** Python library to interact with MySQL.
* **MySQL Database:** Stores structured data.

**7. Methodology and Development Process**

The project will largely follow an iterative and incremental development approach, drawing principles from Agile methodologies, particularly suitable for academic projects requiring adaptability and continuous feedback.

**7.1. Development Approach**

* **Iterative Development:** The project will be broken down into smaller iterations (sprints), with each iteration focusing on delivering a subset of working features.
* **Modular Design:** Emphasis on creating loosely coupled, highly cohesive modules for both frontend and backend to facilitate parallel development and easier maintenance.
* **Version Control:**

7.2. Project Phases

The development will proceed through the following phases:

* **Phase 1: Requirements Analysis & Detailed Design (Current)**
  + Refining user stories and functional requirements.
  + Designing the database schema in detail.
  + Designing API endpoints and data models.
  + Prototyping key UI screens.
* **Phase 2: Backend Development & API Implementation**
  + Setting up the Flask environment and MySQL database.
  + Implementing user authentication (basic).
  + Developing RESTful APIs for expenses, categories, and budgets.
  + Implementing business logic for data validation and calculations.
* **Phase 3: Frontend Development (KivyMD)**
  + Building the mobile application interface.
  + Integrating the KivyMD frontend with the Flask backend via API calls.
  + Implementing core UI functionalities like expense logging, category management, and budget tracking.
* **Phase 4: Reporting & Visualization Implementation**
  + Developing the logic for data aggregation and report generation in Flask.
  + Designing and implementing charts and graphs within the KivyMD application.
* **Phase 5: Testing & Debugging**
  + **Unit Testing:** Testing individual functions and components of both frontend and backend.
  + **Integration Testing:** Verifying communication and data flow between KivyMD, Flask, and MySQL.
  + **System Testing:** End-to-end testing of the entire application.
  + **User Acceptance Testing (UAT):** Gathering feedback from potential users to validate functionality and usability.
* **Phase 6: Deployment & Documentation**
  + Packaging the KivyMD application for mobile platforms (e.g., Android APK).
  + Documenting the project code, architecture, and user manual for academic submission.

**8. System Requirements**

To ensure optimal performance and functionality, the "Expense Tracker" will have the following system requirements:

**8.1. Software Requirements**

* **Operating System:** Android (Minimum Android 7.0 "Nougat" for mobile app testing), Windows/macOS/Linux (for development environment).
* **Programming Languages:** Python 3.8+
* **Frontend Framework:** KivyMD 0.104.2+ (requires Kivy 2.0+)
* **Backend Framework:** Flask 2.0+
* **Database:** MySQL 8.0+
* **Python Libraries (Backend):** mysql-connector-python or SQLAlchemy, Flask-CORS (for cross-origin requests).
* **Development Tools:** Git, VS Code/PyCharm, MySQL Workbench.

**8.2. Hardware Requirements**

* **Development Machine:**
  + Processor: Intel Core i5 or equivalent (or better).
  + RAM: 8 GB (16 GB recommended).
  + Storage: 500 GB SSD (or more).
* **Mobile Device (for testing):**
  + RAM: 2 GB (4 GB recommended).
  + Storage: Minimum 100 MB free space.
  + Internet Connectivity: Required during development and for initial backend setup, but core app will function offline for data logging.

**9. Conclusion**

The "Expense Tracker" project represents a focused effort to leverage modern software development methodologies and a robust technology stack (Python, KivyMD, Flask, MySQL) to address a tangible and prevalent financial management challenge within the middle-class and competitive examination candidate demographics in India. By providing an intuitive, feature-rich, and reliable platform for expense tracking and analysis, the application aims to significantly enhance financial literacy, promote disciplined spending habits, and ultimately contribute to the economic well-being and aspirational fulfillment of its users. This project not only serves as a practical solution but also as a comprehensive academic exercise, demonstrating proficiency in full-stack development principles and user-centric design. The detailed planning outlined in this document sets a clear roadmap for the successful realization of this impactful application.

**10. References (Optional)**

* [If you referred to any specific financial reports, market research, or academic papers about financial literacy or expense tracking, list them here. Example:]
  + "Report on Household Income and Expenditure in India," National Sample Survey Office.
  + [Any specific KivyMD/Flask documentation that significantly influenced design decisions.]