DBMS PROJECT

PROJECT REPORT

**Group Members:-**

Anushka Dwivedi(IIT2022062)  
Kovid Juyal(IIT2022063)  
Priyanshi Khataniya (IIT2022064)  
Pradip Yadav(IIT2022065)  
Saurav Gitte(IIT2022066)

**TABLE OF CONTENT**

1) Introduction……………………………………………………………3

1.2) Problem Statement …………………………………………….3  
1.3)Objectives…………………………………………………………….3

2) Functions………………………………………….………………………..4

3) Tech Stack used:

a) Front End……………………………………………………………5

b) Back End ……………………………………………………………5

4) E-R Diagram……………………………………………………………7

5) Database Schema ………………………………………………..10

6) Keys and Constraints…………………………………………….12

a) Primary Keys

b) Foreign Keys

c) Constraints

7) Project Illustrations………………………………………………14

**EXPENSE TRACKER**

1. **Introduction :**

The Expense Management System is a comprehensive platform designed to simplify financial tracking and empower users to manage their expenses effectively. Our mission is to revolutionize financial awareness by providing a seamless experience for recording income and expenses.

**1.2) Problem Statement :**

Many individuals struggle with effectively managing their personal finances, often leading to overspending, budget deficits, and financial stress. Traditional methods of expense tracking, such as manual record-keeping or spreadsheet management, can be time-consuming and prone to errors. Additionally, users may find it challenging to maintain a comprehensive overview of their expenses and budgets across various categories.

**1.3) Objectives :**

**1. Effortless Expense Tracking**

* Eliminate manual tracking by creating an intuitive platform.
* Capture all transactions effortlessly, from morning coffees to monthly bill payments.

**2. Intuitive Spend Categorization**

* Simplify expense categorization.
* Users can easily classify spending, gaining clarity on essential bills and discretionary expenses.

**3. Actionable Financial Insights**

* Leverage data visualization to generate insightful reports.
* Unveil spending patterns and trends.
* Identify areas for potential savings and optimization.

**2) Functions :**

**1.User Registration:**

* Allows users to create accounts and login with their respective credentials.

**2.Expense Recording:**

* Users can add their expenses.
* Capture transaction details, including date, amount, and category and description.

**3.Expense Categorization:**

* Automatically categorizes expenses based on predefined rules.
* Users can manually adjust new categories if needed.

**4. Budget Allocation:**

* The "Budget" section enables users to allocate budgets to different expense categories.
* Users can specify the expense type and allocate a budget amount for each category using a simple form interface.
* The bar graph shows negative expenditure in case the set budget exceeds.

**5.Data Visualization and Reports:**

* Generates visual reports:
* Gives Expense Summary, Breakdown of expenses by category.
* Trends Over Time: Graphical representation of spending patterns.
* Budget vs. Actual: Compare planned vs. actual expenses.
* Budget Setting and Tracking:Users can set monthly budgets for different categories.

**3) Tech Stack Used:**

**Frontend: HTML, CSS**

* HTML and CSS are fundamental for designing the layout, structure, and visual presentation of your website. They allow you to create an intuitive and visually appealing interface for users to interact with.
* It enables you to make your website responsive by using techniques like media queries, flexible grids, and fluid layouts, ensuring that your Expense Tracker site looks good and functions well on desktops, laptops, tablets, and smartphones.

**Backend: PHP**

* It is essential for the handling business logic, database operations, user authentication, and session management.
* It interacts with the MySQL database to store and retrieve expense data, user information, and budget details.
* PHP processes user requests, generates dynamic content for the frontend, and ensures secure data handling. It facilitates the integration of frontend and backend functionalities, enabling a seamless and interactive user experience on the Expense Tracker website.

**Web Server: APACHE**

* Apache serves as the web server for your Expense Tracker project, delivering HTML, CSS, and JavaScript files to users' browsers. It hosts and serves static frontend files, handles URL routing for navigation, and provides security features to protect frontend resources.
* Apache ensures reliable access to the frontend components, facilitating a seamless user experience for navigating, interacting with, and accessing your Expense Tracker website.

**Database server: MySQL**

* MySQL serves as the backend database server for your Expense Tracker project, storing and managing data related to users, expenses, expense types, and budgets. It provides a robust and scalable platform for efficient data storage, retrieval, and manipulation.
* MySQL ensures data integrity and consistency, enabling reliable tracking and analysis of expenses. Its relational database model facilitates structured organization and retrieval of expense-related information, supporting the functionality and usability of your Expense Tracker website.

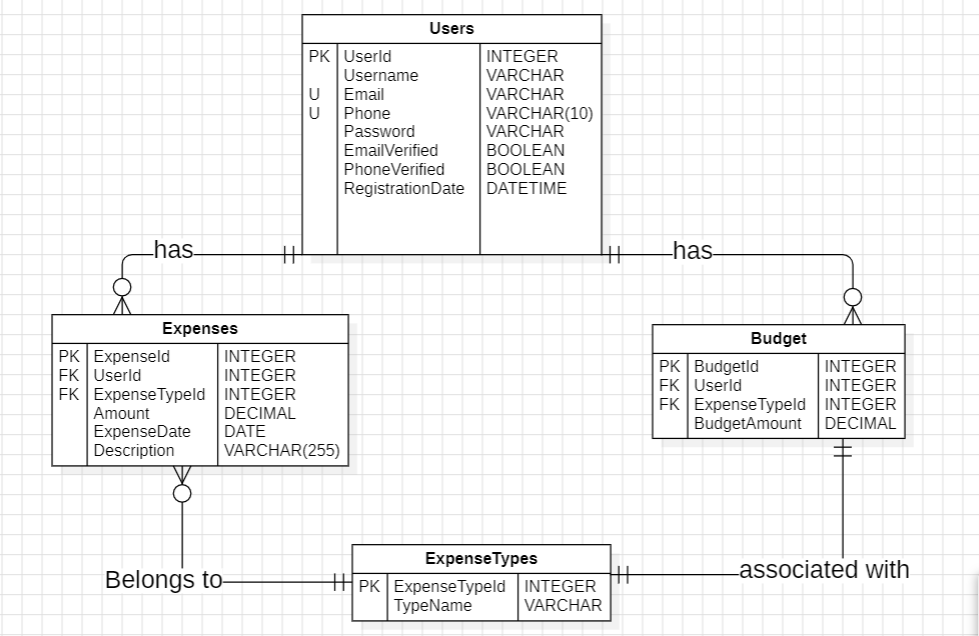
**Web Browser: Google Chrome or any Compatible Browser**

* The frontend is optimized for all modern web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, ensuring wide accessibility and consistent user experience across different platforms.

**Operating System: Windows or any equivalent OS**

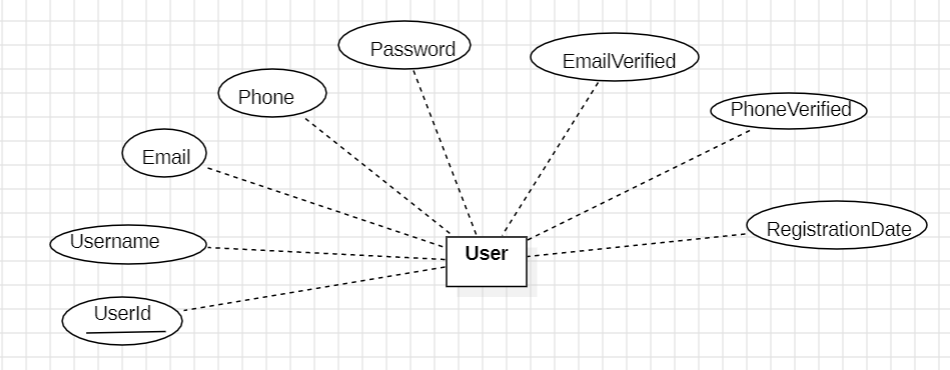
* • The system is designed to be operating system agnostic, functioning seamlessly on Windows, macOS, Linux, or any other system.

**4) E-R Diagram:**

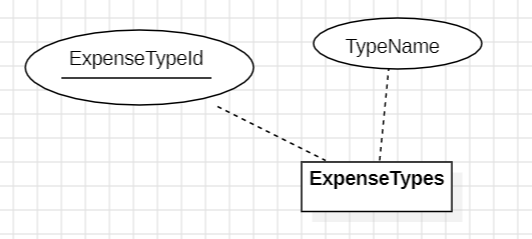
These are tables and relations. This is relational database diagram.  


Now E-R Diagram for various components “

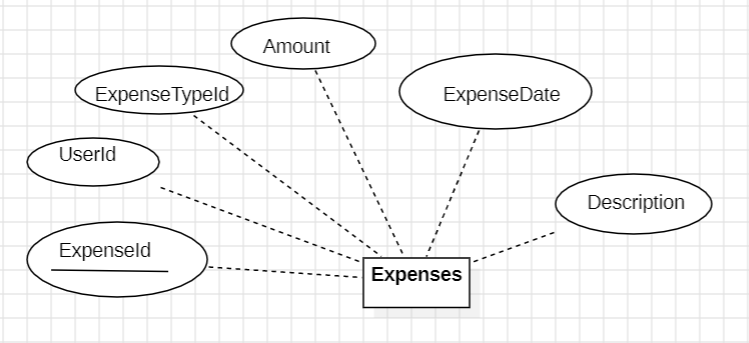
**E-R Diagram for User:**



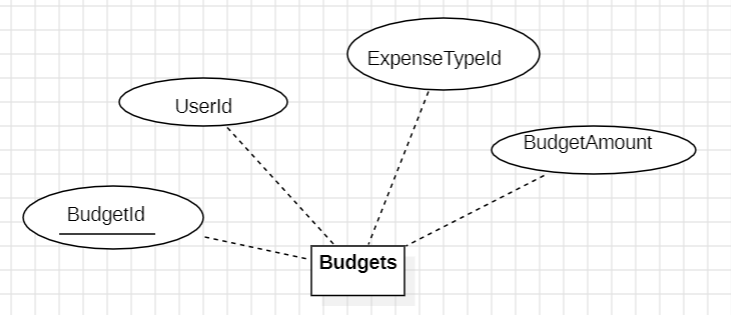
**E-R Diagram for Expense Types:**



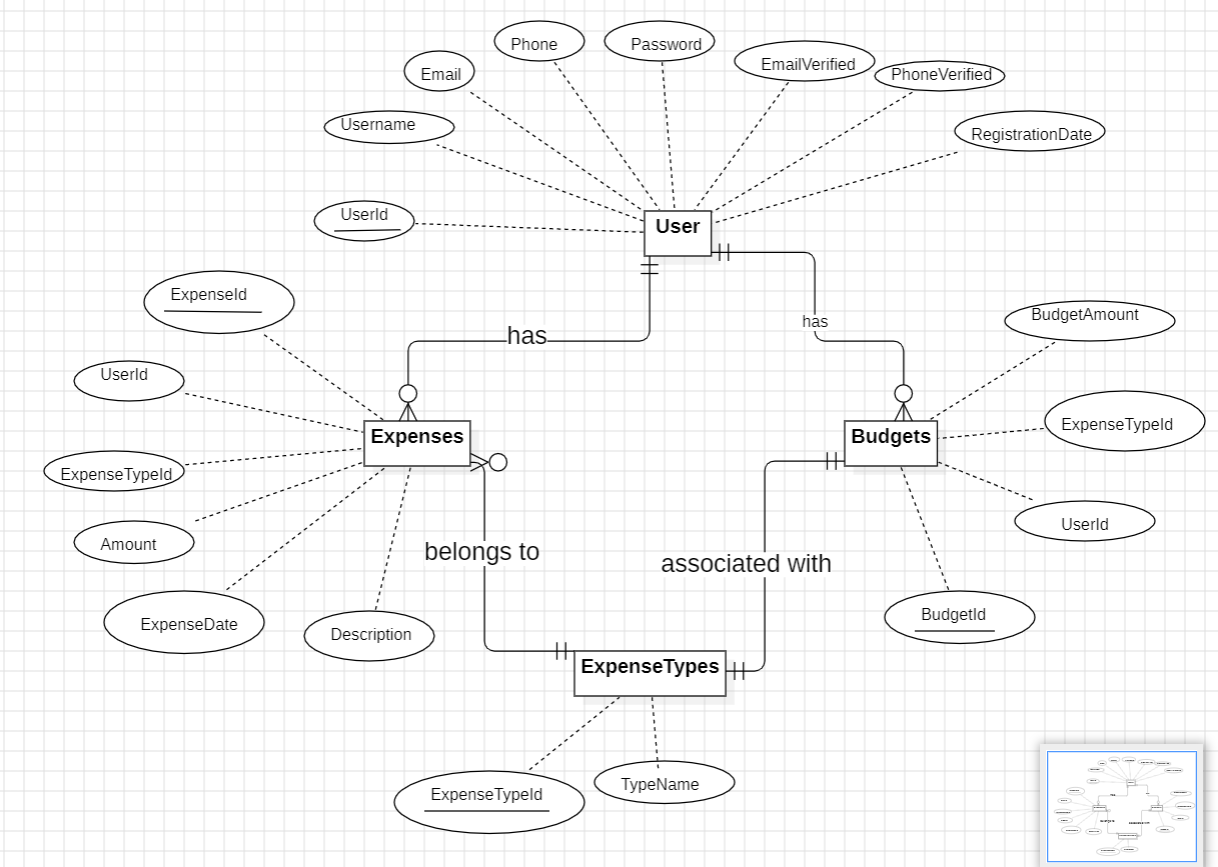
**E-R Diagram for Expenses:**



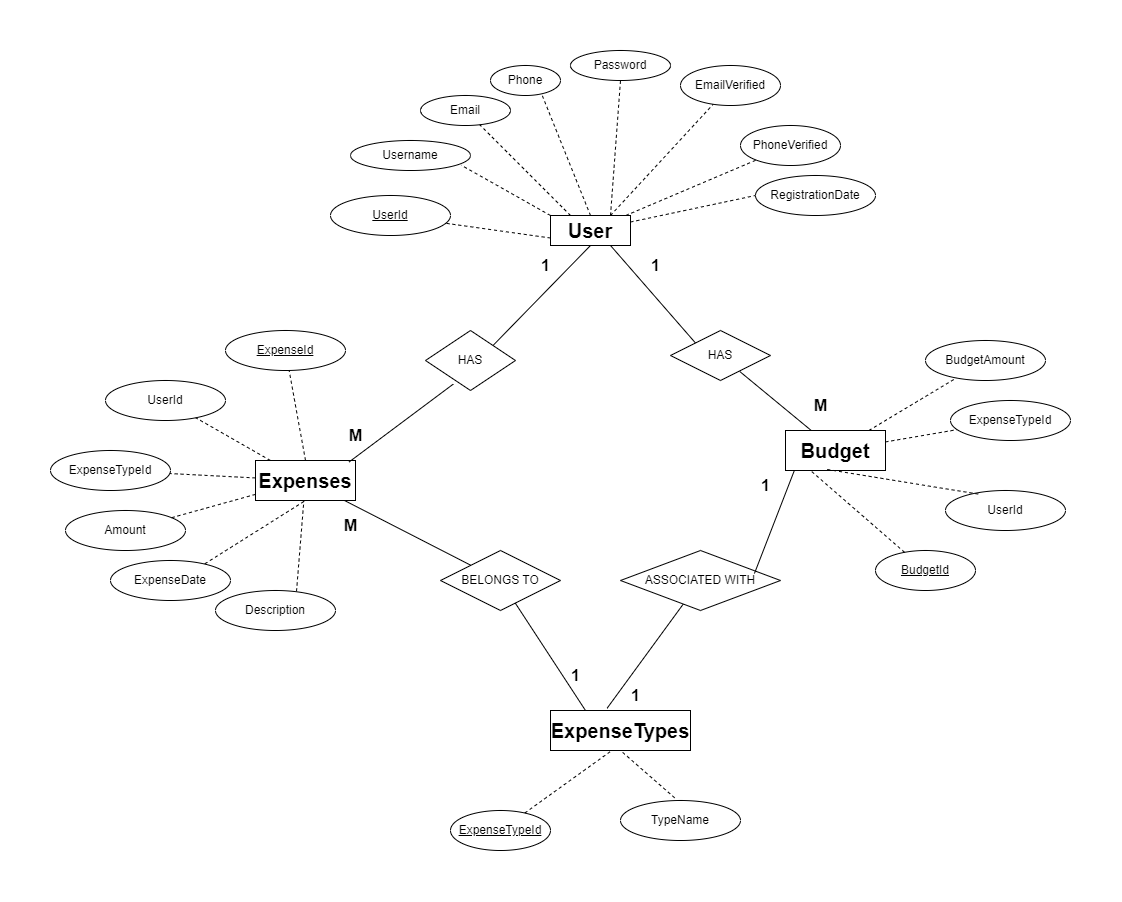
**E-R Diagram for Budgets:**



**Overall E-R diagram ( Made in STARUML )**

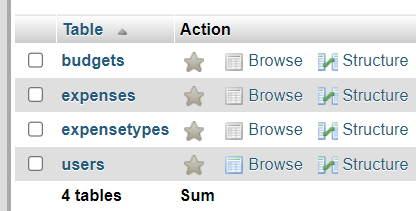


**Overall E-R diagram ( Made in draw.io )**



**5) Database Schemas :**

**4 tables:**

****

**1. User Table**

CREATE TABLE Users (

    UserID INT PRIMARY KEY AUTO\_INCREMENT,

    Username VARCHAR(50) NOT NULL,

    Email VARCHAR(100) UNIQUE NOT NULL,

    Phone VARCHAR(15) UNIQUE NOT NULL,

    Password VARCHAR(100) NOT NULL,

    EmailVerified BOOLEAN DEFAULT FALSE,

    PhoneVerified BOOLEAN DEFAULT FALSE,

    RegistrationDate DATETIME DEFAULT CURRENT\_TIMESTAMP

);

**2.Expense Types Table**

CREATE TABLE ExpenseTypes (

    ExpenseTypeID INT PRIMARY KEY AUTO\_INCREMENT,

    TypeName VARCHAR(50) NOT NULL

);

**3. Expenses Table**

CREATE TABLE Expenses (

    ExpenseID INT PRIMARY KEY AUTO\_INCREMENT,

    UserID INT,

    ExpenseTypeID INT,

    Amount DECIMAL(10, 2) NOT NULL,

    ExpenseDate DATE NOT NULL,

    Description VARCHAR(255),

    FOREIGN KEY (UserID) REFERENCES Users(UserID),

    FOREIGN KEY (ExpenseTypeID) REFERENCES ExpenseTypes(ExpenseTypeID)

);

**4. Budgets Table**

CREATE TABLE Budgets (

    BudgetID INT PRIMARY KEY AUTO\_INCREMENT,

    UserID INT,

    ExpenseTypeID INT,

    BudgetAmount DECIMAL(10, 2) NOT NULL,

    FOREIGN KEY (UserID) REFERENCES Users(UserID),

    FOREIGN KEY (ExpenseTypeID) REFERENCES ExpenseTypes(ExpenseTypeID)

);

**6) Keys and Constraints:**

**a) Primary Keys:**

* Users: UserID
* ExpenseTypes: ExpenseTypeID
* Expenses: ExpenseID
* Budgets: BudgetID

**b) Foreign Keys:**

* Expenses: UserID (References Users(UserID))
* Expenses: ExpenseTypeID (References ExpenseTypes(ExpenseTypeID))
* Budgets: UserID (References Users(UserID))
* Budgets: ExpenseTypeID (References ExpenseTypes(ExpenseTypeID))

**c) Unique Keys:**

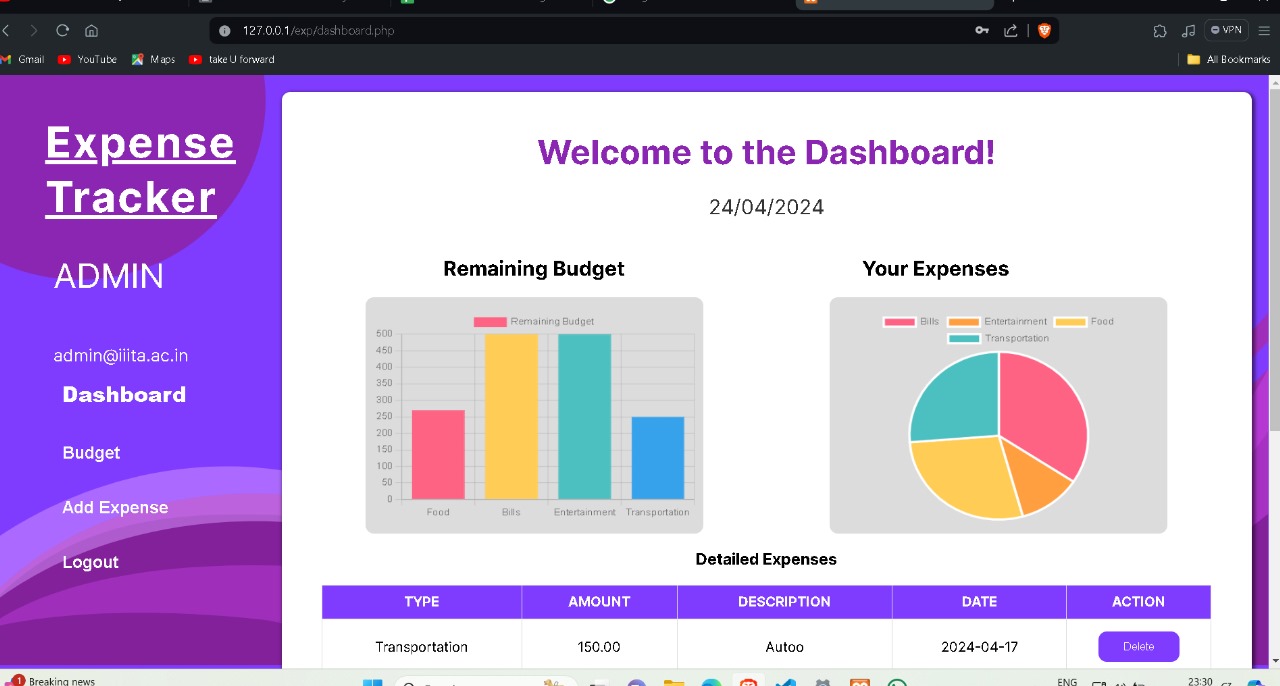
* Users: Email (Unique constraint ensures each email is unique)
* Users: Phone (Unique constraint ensures each phone number is unique)
* ExpenseTypes: TypeName (Unique constraint ensures each expense type name is unique)

**d) Not Null Constraints:**

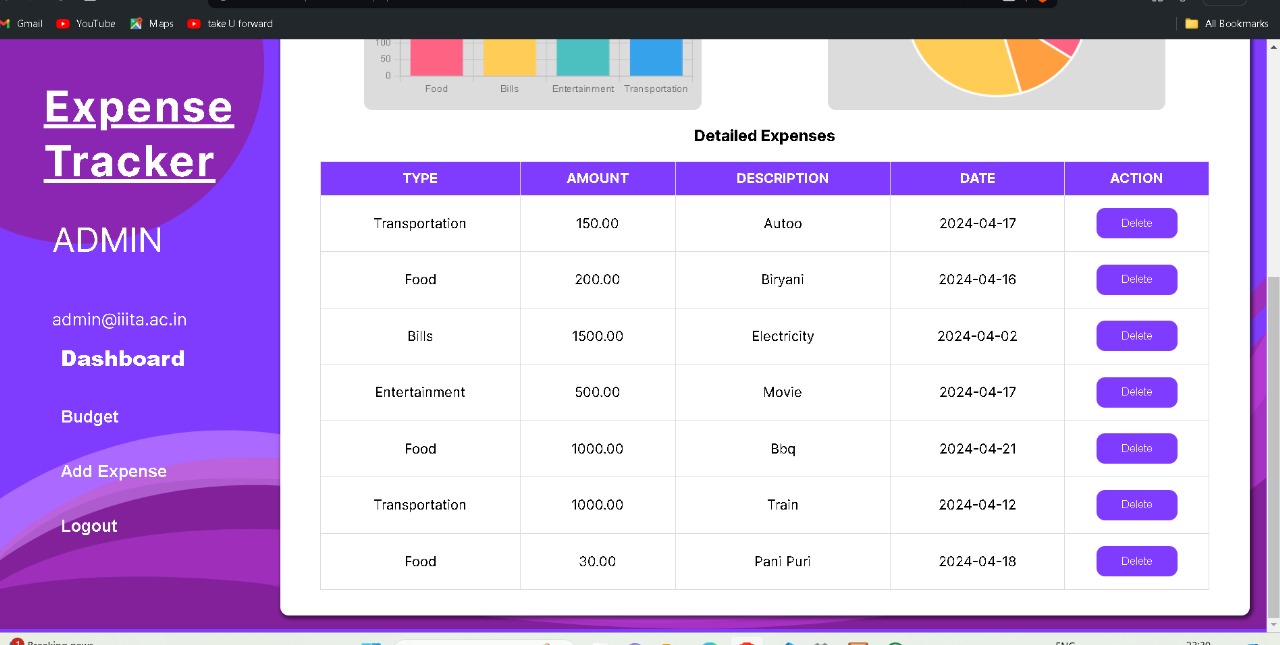
* Users: Username, Email, Phone, Password (Not null constraint ensures these fields are always populated)
* Expenses: Amount, ExpenseDate (Not null constraint ensures these fields are always populated)
* Budgets: BudgetAmount (Not null constraint ensures this field is always populated)

**7) Project Illustrations**

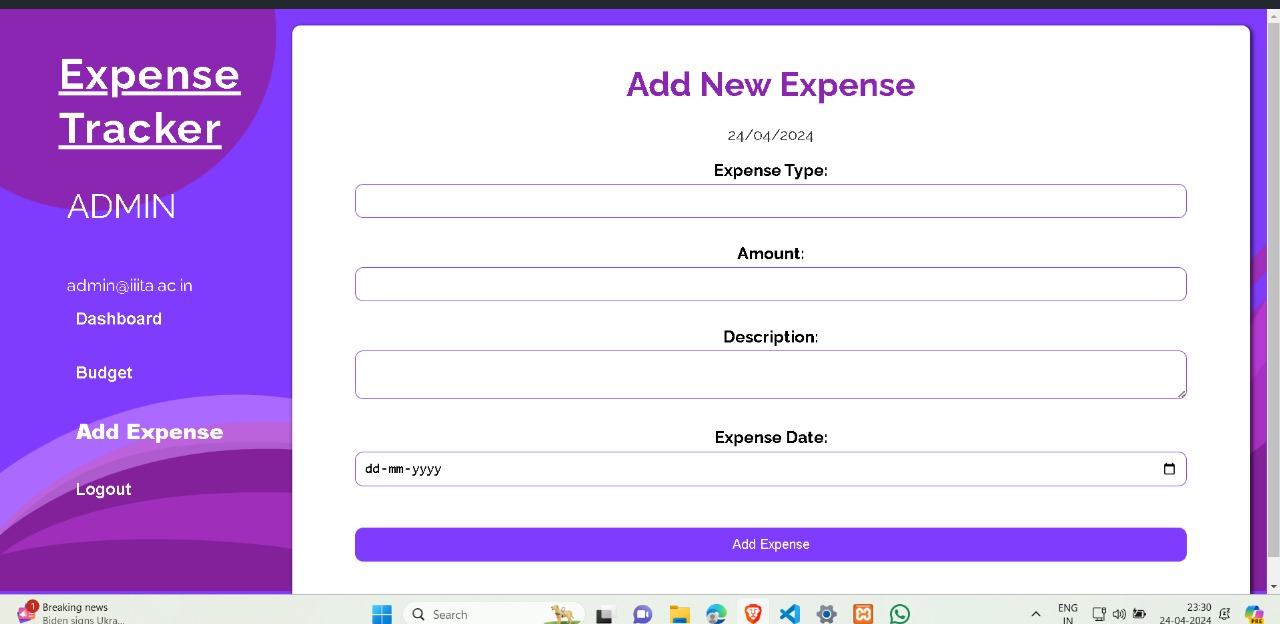
**Dashboard page: (Top)**



**Dashboard page: (Bottom)**



**Expense Page:**



**Budget page:**

