Overview

FinanceMaster is a financial management application designed to help users track and manage their finances effectively. The application provides various functionalities such as transaction history, spending habits analysis, and monitoring currency exchange rates and etc.

Project Structure

The project consists of several components, each responsible for different functionalities within the application. Below is an overview of the key components and their purposes:

Main Components

- **main.cpp**
    - The entry point of the application. It initializes the application and displays the main window.
- **mainwindow.ui**
    - The main user interface layout file for the application. Where implemented buttons and design of the app.
- **mainwindow.cpp**
    - The implementation of the main window's functionalities, including handling button clicks and interactions with other components.
- **mainwindow.h**
    - The header file for the main window, declaring its members and functions.

Widgets and Dialogs

start_page.ui
The UI layout file for the start page of the application.

start_page.cpp
Implementation of the start page functionalities, including file upload and initial setup.

start_page.h
The header file for the start page, declaring its members and functions.

cashback.ui
The UI layout file for the cashback dialog.

cashback.cpp
Implementation of the cashback dialog functionalities.

cashback.h
The header file for the cashback dialog, declaring its members and functions.

currency_rate.ui
The UI layout file for the currency rate widget.

currency_rate.cpp
Implementation of the currency rate functionalities, including fetching and displaying currency rates.

currency_rate.h
The header file for the currency rate widget, declaring its members and functions.

listview.ui
The UI layout file for the list view, used to display a list of bank accounts and recent transactions.

listview.cpp
Implementation of the list view functionalities.

listview.h
The header file for the list view, declaring its members and functions.

sortdialog.ui
The UI layout file for the sort dialog, allowing users to sort data based on different criteria.

sortdialog.cpp
Implementation of the sort dialog functionalities.

sortdialog.h
The header file for the sort dialog, declaring its members and functions.

tableview.ui
The UI layout file for the table view, used to display transaction data in a tabular format.

tableview.cpp
Implementation of the table view functionalities, including data loading and display.

tableview.h
The header file for the table view, declaring its members and functions.

Analysis

analysisdialog.ui

The UI layout file for the analysis dialog, used to display analysis of spending habits.

analysisdialog.cpp

Implementation of the analysis dialog functionalities, including generating and displaying line graphs.

analysisdialog.h

The header file for the analysis dialog, declaring its members and functions.

PieChartWidget

The widget with pie chart of expenses grouping by category.

Utility Files

proxymodel.h

A header file for the proxy model, used for data transformation and sorting.

resource.qrc

The resource file containing references to images and other resources used in the application.

Logging

loggingcategory.cpp

Implementation of the logging functionalities, enabling logging of application events to prevent and check for errors and warnings..

Usage

Running the Application

To run the application, build the project using Qt Creator or another compatible IDE. Ensure that all dependencies are installed, including the necessary Qt modules.

Main Features

- **Start Page**
  - Upload CSV files containing transaction data.
  - Initialize the application with the uploaded data.
- **Transaction History**
  - View transaction data in a tabular format.

    - Sort transactions based on different criteria.
- **Cashback**
    - View cashback information in a separate dialog.
- **Currency Rates**
    - Fetch and display current currency exchange rates using an external API.
- **Spending Analysis**
    - View a line graph showing spending habits over time.

Dependencies

Ensure that the following Qt modules are installed:

- Qt Core
- Qt Widgets
- Qt Network
- Qt Charts (for displaying graphs)

Installation

Clone the Repository:
bash
Копировать код
```
git clone <repository_url>
cd FinanceMaster
```

1.
2. **Open the Project in Qt Creator:**
    - Open `FinanceMaster.pro` in Qt Creator.
3. **Build the Project:**
    - Click on the build button in Qt Creator to compile the project.
4. **Run the Application:**
    - Click on the run button in Qt Creator to start the application.

Logging

Logging is implemented to track application events and errors. The log file is saved to the user's computer upon the application’s termination.

Notes

- Ensure that the CSV files have the correct format and column names to avoid errors during data loading and processing.

- The application requires an active internet connection to fetch currency rates from the external API.

Future Enhancements

- Add more analysis features, such as category-wise spending analysis.
- Implement user authentication and data encryption for better security.
- Enhance the UI/UX with more intuitive design elements and user interactions.

This documentation provides an overview of the FinanceMaster application, its structure, and usage. For more detailed information, refer to the source code and inline comments.