

```
import datetime
```

```
# Class to represent an expense entry
```

```
class Expense:
```

```
    def __init__(self, amount, date, category, notes=None, recurring=False):
```

```
        self.amount = amount
```

```
        self.date = date
```

```
        self.category = category
```

```
        self.notes = notes
```

```
        self.recurring = recurring
```

```
# Class to manage expenses and monthly budget
```

```
class ExpenseTracker:
```

```
    def __init__(self):
```

```
        self.expenses = []
```

```
    def add_expense(self, expense):
```

```
        self.expenses.append(expense)
```

```
    def get_total_expenses(self, month=None):
```

```
        if month:
```

```
            total_expenses = sum(expense.amount for expense in self.expenses if  
expense.date.startswith(month))
```

```
        else:
```

```
            total_expenses = sum(expense.amount for expense in self.expenses)
```

```
        return total_expenses
```

```
    def get_expenses_by_category(self, month=None):
```

```
        if month:
```

```
            expenses_by_category = {}
```

```
            for expense in self.expenses:
```

```

        if expense.date.startswith(month):
            expenses_by_category[expense.category] = expenses_by_category.get(expense.category,
0) + expense.amount
            return expenses_by_category
        else:
            expenses_by_category = {}
            for expense in self.expenses:
                expenses_by_category[expense.category] = expenses_by_category.get(expense.category, 0)
+ expense.amount
            return expenses_by_category

```

```

def is_within_budget(self, month_budget, month=None):
    total_expenses = self.get_total_expenses(month)
    return total_expenses <= month_budget

```

Function to prompt user for expense details

```

def get_expense_details():
    amount = float(input("Enter the amount you spent: "))
    date = input("Enter the date of the expense (YYYY-MM-DD): ")
    category = input("Enter the category of the expense: ")
    notes = input("Enter any notes for the expense (optional): ")
    recurring = input("Is this a recurring expense? (yes/no): ").lower() == 'yes'
    return Expense(amount, date, category, notes, recurring)

```

Main function to interact with the user

```

def main():
    print("Welcome to the Expense Tracker App!")
    tracker = ExpenseTracker()
    monthly_budget = float(input("Enter your monthly budget: $"))
    while True:
        print("\n1. Add Expense\n2. View Total Monthly Expenses\n3. View Monthly Budget Status\n4.
Exit")

```

```

choice = input("Enter your choice: ")

if choice == '1':
    expense = get_expense_details()
    tracker.add_expense(expense)
    print("Expense added successfully!")

elif choice == '2':
    month = input("Enter month to view total expenses (YYYY-MM): ")
    total_expenses = tracker.get_total_expenses(month)
    print(f"Total expenses for {month}: ${total_expenses:.2f}")
    expenses_by_category = tracker.get_expenses_by_category(month)
    print("Expenses by category:")
    for category, total_amount in expenses_by_category.items():
        print(f"{category}: ${total_amount:.2f}")

elif choice == '3':
    month = input("Enter month to check budget status (YYYY-MM): ")
    if tracker.is_within_budget(monthly_budget, month):
        print("You are within the monthly budget.")
    else:
        print("You have exceeded the monthly budget.")

elif choice == '4':
    print("Exiting...")
    break

else:
    print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()

```