```
import mysql.connector
import tkinter as tk
from tkinter import messagebox, ttk

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",      # Update with your MySQL host
    user="root",    # Replace with your MySQL username
    password="12345",# Replace with your MySQL password
    database="expense_tracker_db"  # Database name you created
)
c = conn.cursor()

# Function to add an expense to the database
def add_expense():
    category = category_entry.get()
    amount = amount_entry.get()
    date = date_entry.get()
    description = description_entry.get()

    if category and amount and date:
        try:
            amount = float(amount)  # Convert amount to a float
            sql = "INSERT INTO expenses (category, amount, date, description) VALUES (%s, %s, %s, %s)"
            values = (category, amount, date, description)
            c.execute(sql, values)
            conn.commit()
            messagebox.showinfo("Success", "Expense added successfully!")
            clear_entries()
            view_expenses()  # Refresh the expenses table
        except ValueError:
```

```
            messagebox.showerror("Error", "Amount must be a number.")
    else:
        messagebox.showerror("Error", "Please fill in all fields.")

# Function to clear entry fields after adding
def clear_entries():
    category_entry.delete(0, tk.END)
    amount_entry.delete(0, tk.END)
    date_entry.delete(0, tk.END)
    description_entry.delete(0, tk.END)

# Function to view all expenses
def view_expenses():
    # Clear the tree view table
    for i in tree.get_children():
        tree.delete(i)
    c.execute("SELECT * FROM expenses")
    for row in c.fetchall():
        tree.insert("", "end", values=row)

# Function to calculate total expenditure in a date range
def calculate_total():
    start_date = start_date_entry.get()
    end_date = end_date_entry.get()
    c.execute("SELECT SUM(amount) FROM expenses WHERE date BETWEEN %s AND %s", (start_date,
end_date))
    total = c.fetchone()[0] or 0
    messagebox.showinfo("Total Expenditure", f"Total expenditure from {start_date} to {end_date}:
{total}")

# Create the main application window
root = tk.Tk()
```

```
root.title("Personal Expense Tracker")

# Category input
tk.Label(root, text="Category").grid(row=0, column=0, padx=10, pady=5)
category_entry = tk.Entry(root)
category_entry.grid(row=0, column=1, padx=10, pady=5)

# Amount input
tk.Label(root, text="Amount").grid(row=1, column=0, padx=10, pady=5)
amount_entry = tk.Entry(root)
amount_entry.grid(row=1, column=1, padx=10, pady=5)

# Date input
tk.Label(root, text="Date (YYYY-MM-DD)").grid(row=2, column=0, padx=10, pady=5)
date_entry = tk.Entry(root)
date_entry.grid(row=2, column=1, padx=10, pady=5)

# Description input
tk.Label(root, text="Description").grid(row=3, column=0, padx=10, pady=5)
description_entry = tk.Entry(root)
description_entry.grid(row=3, column=1, padx=10, pady=5)

# Add expense button
add_button = tk.Button(root, text="Add Expense", command=add_expense)
add_button.grid(row=4, column=1, padx=10, pady=5)

# Treeview for displaying expenses
columns = ("expense_id", "category", "amount", "date", "description")
tree = ttk.Treeview(root, columns=columns, show="headings")
tree.heading("expense_id", text="ID")
tree.heading("category", text="Category")
```

```
tree.heading("amount", text="Amount")
tree.heading("date", text="Date")
tree.heading("description", text="Description")
tree.grid(row=5, column=0, columnspan=3, padx=10, pady=10)

# View expenses button
view_button = tk.Button(root, text="View Expenses", command=view_expenses)
view_button.grid(row=6, column=0, padx=10, pady=5)

# Total expenditure calculation input fields
tk.Label(root, text="Start Date (YYYY-MM-DD)").grid(row=7, column=0, padx=10, pady=5)
start_date_entry = tk.Entry(root)
start_date_entry.grid(row=7, column=1, padx=10, pady=5)

tk.Label(root, text="End Date (YYYY-MM-DD)").grid(row=8, column=0, padx=10, pady=5)
end_date_entry = tk.Entry(root)
end_date_entry.grid(row=8, column=1, padx=10, pady=5)

# Calculate total button
total_button = tk.Button(root, text="Calculate Total Expenditure", command=calculate_total)
total_button.grid(row=9, column=1, padx=10, pady=5)

# Run the Tkinter main loop
root.mainloop()

# Close the database connection on exit
c.close()
conn.close()
```