```
import json
from datetime import datetime

class BudgetTracker:
    def __init__(self):
        self.categories = {}

    def add_category(self, category_name, initial_balance=0):
        if category_name not in self.categories:
            self.categories[category_name] = {"balance": initial_balance, "expenses": []}
            print(f"{category_name} category created successfully with an initial balance of
${initial_balance}.")
        else:
            print(f"{category_name} category already exists!")

    def add_income(self, category_name, amount):
        if category_name in self.categories:
            self.categories[category_name]["balance"] += amount
            print(f"Income of ${amount} added to {category_name} category.")
        else:
            print(f"{category_name} category does not exist.")

    def add_expense(self, category_name, amount, description=""):
        if category_name in self.categories:
            if self.categories[category_name]["balance"] >= amount:
                self.categories[category_name]["balance"] -= amount
                expense = {"amount": amount, "description": description, "date":
datetime.now().strftime("%Y-%m-%d")}
                self.categories[category_name]["expenses"].append(expense)
                print(f"Expense of ${amount} deducted from {category_name} category.")
            else:
                print(f"Not enough funds in {category_name} category to cover the expense.")
        else:
            print(f"{category_name} category does not exist.")

    def set_budget(self, category_name, budget_amount):
        if category_name in self.categories:
            self.categories[category_name]["budget"] = budget_amount
            print(f"Budget of ${budget_amount} set for {category_name} category.")
        else:
            print(f"{category_name} category does not exist.")

    def view_categories(self):
        print("Categories and their balances:")
        for category, info in self.categories.items():
            print(f"{category}: Balance - ${info['balance']}, Budget - ${info.get('budget', 'Not
set')}")
```

```
    def view_expenses(self, category_name):
        if category_name in self.categories:
            print(f"Expenses for {category_name} category:")
            for expense in self.categories[category_name]["expenses"]:
                print(f"Date: {expense['date']}, Amount: ${expense['amount']}, Description:
{expense['description']}")
        else:
            print(f"{category_name} category does not exist.")

    def save_to_file(self, filename="budget_data.json"):
        with open(filename, "w") as f:
            json.dump(self.categories, f)
        print("Budget data saved successfully.")

    def load_from_file(self, filename="budget_data.json"):
        try:
            with open(filename, "r") as f:
                self.categories = json.load(f)
            print("Budget data loaded successfully.")
        except FileNotFoundError:
            print("File not found. No budget data loaded.")
```