```
import os

DATA_FILE = "students.txt"

# Blueprint for managing individual student data
class StudentInfo:
    def __init__(self, roll_id, name, dept):
        self.roll_id = roll_id
        self.name = name
        self.dept = dept

    def format_record(self):
        return f"{self.roll_id},{self.name},{self.dept}"

    @staticmethod
    def parse_line(data_line):
        elements = data_line.strip().split(',')
        if len(elements) == 3:
            return StudentInfo(*elements)
        else:
            print(f" ⚠️ Unable to interpret line: {data_line.strip()}")
            return None


class RecordsManager:
    def __init__(self, data_path):
        self.data_path = data_path

    def insert_student(self, info_obj):
```

```
    try:
        with open(self.data_path, 'a', encoding='utf-8') as file:
            file.write(info_obj.format_record() + '\n')
        print("✅ Entry successfully added.\n")
    except Exception as err:
        print(f"❌ Write operation failed: {err}")

def show_all(self):
    if not os.path.exists(self.data_path):
        print("📂 No student entries available.\n")
        return
    print("\n📋 Complete Student List:\n")
    try:
        with open(self.data_path, 'r', encoding='utf-8') as file:
            for entry in file:
                info = StudentInfo.parse_line(entry)
                if info:
                    print(f"Roll No: {info.roll_id} | Name: {info.name} | Department: {info.dept}")
    except Exception as err:
        print(f"❌ Failed to read file: {err}")
    print()

def search_student(self, roll_id):
    try:
        with open(self.data_path, 'r', encoding='utf-8') as file:
            for entry in file:
                info = StudentInfo.parse_line(entry)
                if info and info.roll_id == roll_id:
                    print(f"\n🔍 Found ➤ Roll: {info.roll_id} | Name: {info.name} | Dept: {info.dept}\n")
```

```
                return
        print(f"🔎 No match for Roll Number: {roll_id}")
    except Exception as err:
        print(f"❌ File error: {err}")

def modify_student(self, roll_id):
    was_updated = False
    entries = []
    try:
        with open(self.data_path, 'r', encoding='utf-8') as file:
            for entry in file:
                info = StudentInfo.parse_line(entry)
                if info and info.roll_id == roll_id:
                    print(f"Current ➤ {info.format_record()}")
                    info.name = input("New Name: ")
                    info.dept = input("New Department: ")
                    was_updated = True
                entries.append(info.format_record() + "\n")

        if was_updated:
            with open(self.data_path, 'w', encoding='utf-8') as file:
                file.writelines(entries)
            print("✅ Update completed.\n")
        else:
            print("⚠️ No such roll number in records.\n")
    except Exception as err:
        print(f"❌ Update error: {err}")

def remove_student(self, roll_id):
```

```
        found = False
        updated_list = []
        try:
            with open(self.data_path, 'r', encoding='utf-8') as file:
                for entry in file:
                    info = StudentInfo.parse_line(entry)
                    if info and info.roll_id != roll_id:
                        updated_list.append(info.format_record() + "\n")
                    else:
                        found = True

            if found:
                with open(self.data_path, 'w', encoding='utf-8') as file:
                    file.writelines(updated_list)
                print("🗑️ Entry deleted.\n")
            else:
                print("⚠️ No matching roll number found.\n")
        except Exception as err:
            print(f"❌ Error removing entry: {err}")


def launch_app():
    manager = RecordsManager(DATA_FILE)
    while True:
        print("\n🎓 Student Info Manager")
        print("1. Add New Entry")
        print("2. View All Students")
        print("3. Search by Roll Number")
        print("4. Edit Student Info")
```

```
print("5. Delete Entry")
print("6. Exit")

option = input("Choose option (1-6): ").strip()

if option == '1':
    roll = input("Enter Roll Number: ").strip()
    name = input("Enter Full Name: ").strip()
    dept = input("Enter Department: ").strip()
    new_entry = StudentInfo(roll, name, dept)
    manager.insert_student(new_entry)

elif option == '2':
    manager.show_all()

elif option == '3':
    roll = input("Roll Number to Search: ").strip()
    manager.search_student(roll)

elif option == '4':
    roll = input("Roll Number to Update: ").strip()
    manager.modify_student(roll)

elif option == '5':
    roll = input("Roll Number to Delete: ").strip()
    manager.remove_student(roll)

elif option == '6':
    print(" 👋 Exiting program.")
```

```
            break


        else:
            print(" ❗ Please enter a valid choice (1-6).\n")



if __name__ == "__main__":
    launch_app()
```