

Test cases & execution evidence

Fictional student details are used below. Input lines appear in order, separated by /. The expected column shows key output rather than the repeated menu. Run these cases in order, starting without students.json.

Executed verification

Python 3.10.12; isolated temporary data; 18/18 checks passed. Python 3.8 grammar check passed too.

Case	Input / setup	Expected result
01 Empty view	2 / 6	No student records found.
02 Add	1 / bca001 / Asha Rao / 20 / BCA / asha@example.com / 6	Student added successfully. ID stored as BCA001.
03 Duplicate	1 / BCA001 / 6	That student ID already exists.
04 Restart + view	2 / 6	Student Id: BCA001; Name: Asha Rao; Age: 20; Total records: 1
05 Search	3 / bca001 / 3 / ASHA / 6	Both searches show Asha Rao.
06 Update	4 / bca001 / [blank] / 21 / [blank] / [blank] / 6	Student updated successfully. Other fields retained.
07 Persistence	Restart: 2 / 6	Age: 21
08 Cancel delete	5 / bca001 / no / 6	Deletion cancelled.
09 Missing	4 / missing / 5 / missing / 3 / missing / 6	Update/delete: Student ID not found. Search: No student records found.
10 Bad menu/search	9 / 3 / [blank] / 6	Invalid menu choice. Enter a number from 1 to 6. / Search cannot be empty.
11 Validation	1 / bad id / bca002 / 123 / Ravi Shah / x / 14 / 101 / 19 / X / BCA / not-email / ravi@example.com / 6	Invalid ID, name, age, course and email rejected; valid replacements accepted. Student added.
12 Delete	5 / bca001 / YES / 6	Student deleted successfully.
13 Gone on restart	3 / bca001 / 6	No student records found.
14 Damaged file	Write {broken to students.json; run app	Cannot load data safely. Exit 1; original bytes unchanged.
15 Save failure	Mock save_records() to raise OSError; call commit()	Save failed; no changes committed. Returns False; memory unchanged.
16 File duplicate	Load a JSON list with two identical valid records	ValueError: Data file contains duplicate student IDs.
17 Wrong shape	Load JSON object {}	ValueError: Data file must contain a JSON list.
18 Incomplete add	Send 1 / BCA003; close standard input	Session ended. Earlier committed changes are saved. No data file created.

Sample console run

Captured from two real runs on 13 October 2026. Typed input appears after each prompt; the repeated menu and the full listings between actions are trimmed. Session B reopens the program, so students.json from session A is loaded.

```
$ python3 student_records.py
Student Record Management System
Choose 1-6: 1
Student ID: bca001
Name: Asha Rao
Age: 20
Course: BCA
Email: asha@example.com
Student added successfully.
Choose 1-6: 1
Student ID: bca002
Name: Ravi Shah
Age: 19
Course: BCA
Email: ravi@example.com
Student added successfully.
Choose 1-6: 6
Goodbye. All committed changes are saved.

$ python3 student_records.py
Choose 1-6: 3
Search by exact ID or part of name: asha
-----
Student Id: BCA001
Name: Asha Rao
Age: 20
Course: BCA
Email: asha@example.com
-----
Total records: 1
Choose 1-6: 4
Student ID to update: bca002
ID stays the same. Press Enter to keep any other field.
Name [Ravi Shah; Enter to keep]:
Age [19; Enter to keep]: 21
Course [BCA; Enter to keep]:
Email [ravi@example.com; Enter to keep]:
Student updated successfully.
Choose 1-6: 5
Student ID to delete: bca001
Delete BCA001 (Asha Rao)? Type YES: YES
Student deleted successfully.
Choose 1-6: 6
Goodbye. All committed changes are saved.
```

What was checked

Cases 01-14 and 18 ran the script through subprocess input. Cases 15-17 checked storage functions directly. Assertions checked the output, exit status and saved data where relevant. A passing test covers its listed situation, not every possible failure.