

Flight Reservation Manager — Assignment

Objective

Build a console-based Flight Reservation Manager.

This app will let airline staff create, view, update, and delete flight reservations from the command line.

All data must be stored in a file (reservations.txt) so it persists between program runs.

Instructions

- Build a console application using your language of choice (Node.js or C#).
- Use file I/O to persist data in a simple text-based format.
- Handle invalid input gracefully — don't crash.
- Data file: **reservations.txt**. Start empty if no file exists yet.

Run the app using your usual method:

- For Node.js: `node app.js`
- For C#: Compile and run your .exe

Menu Design & Flow

When the app starts, show a main menu like this:

```
=====
      Flight Reservation Manager
=====
1. Create new reservation
2. View all reservations
3. Update a reservation
4. Cancel a reservation
5. Exit
=====
Select an option (1-5):
```

If the user enters an invalid option, display:

```
Invalid option. Please enter a number between 1 and 5.
```

Then redisplay the main menu.

Reservation Data Model

Each reservation record in your data file should be stored as a single line with fields separated by pipes (|):

```
1|John Doe|AI202|2025-08-01|12A|Confirmed
2|Jane Smith|UA156|2025-08-15|8B|Confirmed
3|Bob Johnson|DL789|2025-09-02|15C|Cancelled
```

Fields (in order):

- **id** — Unique integer ID (incremental, starting from 1)
- **passengerName** — Full name
- **flightNumber** — e.g., AI202
- **date** — YYYY-MM-DD format
- **seatNumber** — e.g., 12A
- **status** — Confirmed / Cancelled

Menu Options

1. Create New Reservation

Prompt the user for:

- Passenger name
- Flight number
- Date of flight (validate YYYY-MM-DD format)
- Seat number

Set status to "Confirmed" by default.

Assign a unique id (find the highest existing ID + 1).

Save to file and display:

```
Reservation created successfully with ID: X
```

Then return to main menu.

2. View All Reservations

Load and display all reservations in a nice table format:

ID	Passenger Name	Flight	Date	Seat	Status
1	John Doe	AI202	2025-08-01	12A	Confirmed
2	Jane Smith	UA156	2025-08-15	8B	Confirmed
3	Bob Johnson	DL789	2025-09-02	15C	Cancelled

If no reservations exist:

```
No reservations found.
```

3. Update a Reservation

Ask user for Reservation ID to update.

If found, prompt for each field and allow editing. (Press Enter to keep old value.)

Save changes to file.

Reservation updated successfully.

If not found:

Reservation with ID X not found.

4. Cancel a Reservation

Ask user for Reservation ID.

If found, set status to "Cancelled" and save.

Reservation cancelled successfully.

If not found:

Reservation with ID X not found.

5. Exit

Display: **Goodbye!** and exit the program.

Technical Notes

Text File Format Details

- Each reservation occupies one line in the text file
- Fields are separated by the pipe character (|)
- No headers in the file - just data rows
- Empty file means no reservations exist
- When updating/deleting, read all lines, modify in memory, then rewrite the entire file

Data Validation

- Ensure date follows YYYY-MM-DD format
- Prevent duplicate seat numbers on the same flight and date
- Handle file I/O errors gracefully
- Validate that reservation IDs are numeric when searching/updating

Example File Content (reservations.txt):

```
1|John Doe|AI202|2025-08-01|12A|Confirmed
2|Jane Smith|UA156|2025-08-15|8B|Confirmed
3|Bob Johnson|DL789|2025-09-02|15C|Cancelled
```