# AI ASSISTED CODING

## LAB TEST-2

**SRIMANI**

**2403A51275**

**BATCH-12**

### **SET-J**

**J.1 — [S18J1] Parse INI config**

**PROMPT-**

Parse INI-style config text into a nested Python dict[str, dict[str, object]].  
Trim whitespace.  
Cast numeric strings like "5432" to int.

**CODE-**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

**CODE EXPLANATION-**

* **Read the text line by line**

Break the INI text into lines.

Clean spaces from start and end.

* **Skip useless lines**

Empty lines.

Comment lines starting with #.

* **Handle section headers**

If a line looks like [section], take the name inside brackets.

Add it as a **new dictionary** in the main result.

* **Handle key=value pairs**

Split the line at the first =.

Trim spaces around key and value

* **Convert numeric values**

If value is digits only (like "5432") → convert to integer (5432).

If it’s a negative number string (like "-10") → also convert to integer (-10).

Otherwise, keep as a string.

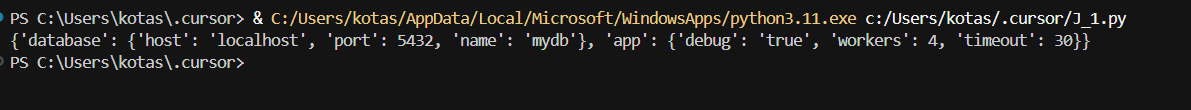
* **Store inside the current section**

Save the key-value pair inside the section’s dictionary.

* **Return the full dictionary**

After all lines are processed, return the nested dictionary.

**OUTPUT-**



**J.2 — [S18J2] Compute average SLA response time**

**PROMPT-**

Write Python code to compute the average ticket duration in minutes from a list of dicts containing opened/closed ISO timestamps. Return the integer average.

**CODE-**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

CODE EXPLANATION-

* The function takes a list of tickets. Each ticket has an opened and closed timestamp.
* It creates an empty list to store durations in minutes.
* For every ticket:

Get the opened and closed time.

If any is missing, skip it.

* Convert the ISO timestamp strings into Python datetime objects.
* Subtract opened time from closed time → this gives the time difference.
* Convert the difference into minutes and store it in the list.
* If timestamps are invalid, skip that ticket.
* After all tickets:
* If the list is empty, return 0.
* Otherwise, take the average of durations and return it as an integer.
* In the example: durations are **90, 105, 75 minutes** → average is **90 minutes.**

**OUTPUT-**

