Here is the corrected code:

utils/logger.py

import logging

import os

import json

def setup\_logger(config\_path):

try:

with open(config\_path, 'r') as f:

config = json.load(f)

except (FileNotFoundError, json.JSONDecodeError) as e:

logging.basicConfig(level=logging.INFO)

logging.error(f"Error loading logging configuration: {e}")

return

```log\_config = config.get('logging', {})

log\_level = log\_config.get('level', 'INFO').upper()

log\_file = log\_config.get('file', 'riscv\_pipeline\_manager.log')

numeric\_level = getattr(logging, log\_level, logging.INFO)

logging.basicConfig(

level=numeric\_level,

format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',

handlers=[

logging.FileHandler(log\_file),

logging.StreamHandler()

]

)

```

This code now properly handles cases where the logging configuration file does not exist or cannot be parsed as JSON. If an error occurs while loading the configuration file, the code sets up a basic logging configuration and logs an error message.