**Assignment**

**CSA0805 – Python Programming**

|  |  |
| --- | --- |
| **Register Number** | **192325140** |
| **Name** | **M.Reddy Jayasree** |

**Title:**

**Config File Parser**:

**Problem Statement:**

Create a Python program that parses configuration files in INI or YAML format using the **configparser** or **pyyaml** module, respectively, and extracts key-value pairs for use in other applications or scripts

**Code:**

import configparser

import yaml

import os

def parse\_ini(file\_path):

config = configparser.ConfigParser()

config.read(file\_path)

data = {}

for section in config.sections():

data[section] = {}

for key in config[section]:

data[section][key] = config[section][key]

return data

def parse\_yaml(file\_path):

with open(file\_path, 'r') as file:

data = yaml.safe\_load(file)

return data

def parse\_config(file\_path):

if not os.path.exists(file\_path):

raise FileNotFoundError(f"The file {file\_path} does not exist.")

\_, file\_extension = os.path.splitext(file\_path)

if file\_extension.lower() == '.ini':

return parse\_ini(file\_path)

elif file\_extension.lower() == '.yaml' or file\_extension.lower() == '.yml':

return parse\_yaml(file\_path)

else:

raise ValueError("Unsupported file format. Please provide an INI or YAML file.")

if \_\_name\_\_ == "\_\_main\_\_":

# Example usage

file\_path = 'config.yaml' # Replace with your file path

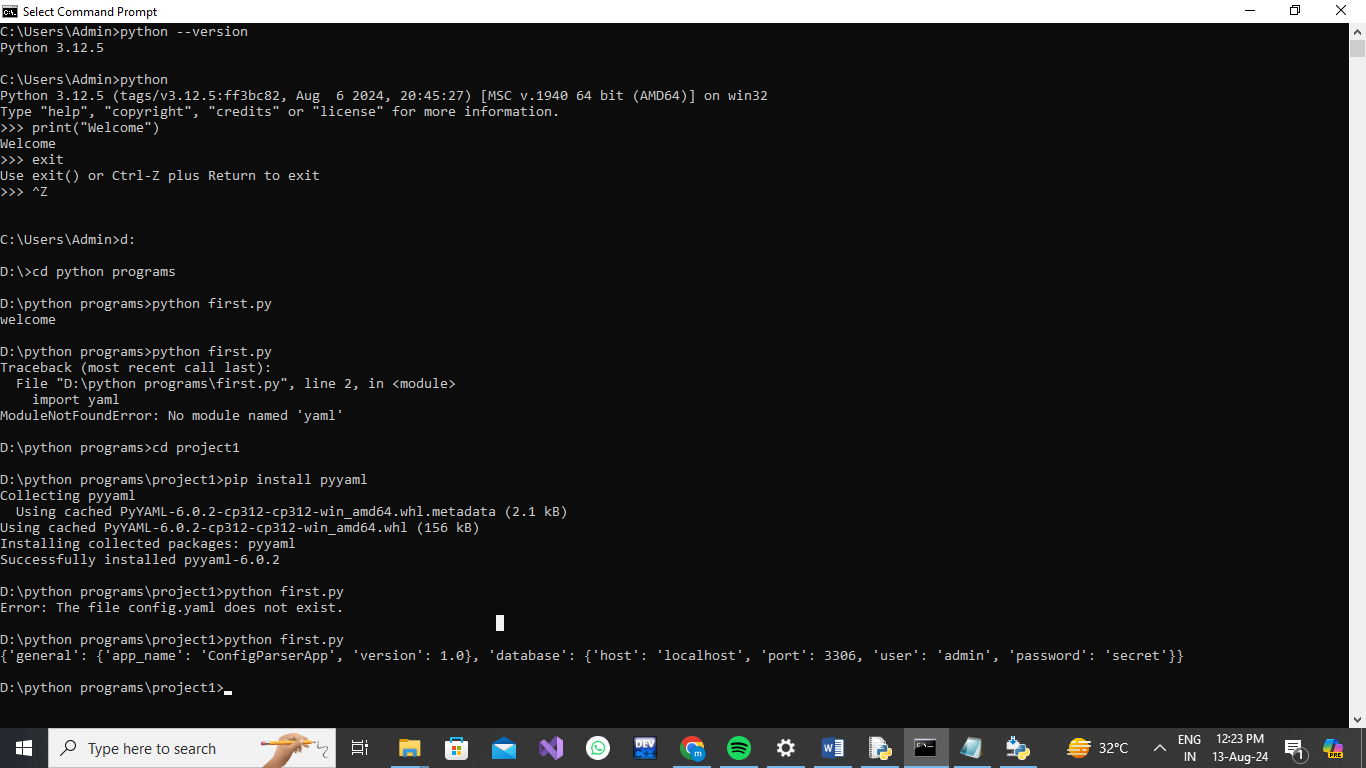
try:

config\_data = parse\_config(file\_path)

print(config\_data)

except (FileNotFoundError, ValueError, yaml.YAMLError, configparser.Error) as e:

print(f"Error: {e}")

**Output Screen Shots:**

**Conclusion:**

This Python program efficiently handles the parsing of configuration files in both INI and YAML formats, using the **configparser** and **pyyaml** modules, respectively. It reads these files, extracts key-value pairs, and provides a flattened dictionary of configurations suitable for use in various applications or scripts. This versatility allows for the integration of configuration management into different systems, making it easier to adapt and maintain configurations in a consistent format. Whether you are dealing with simple INI files or complex YAML structures, this approach ensures that your configuration data is accessible and manageable.