**Assignment**

**CSA0805 – Python Programming**

|  |  |
| --- | --- |
| **Register Number** | **192324224** |
| **Name** | **Kadiyala. showkath ali** |

**Title: :Directory Tree Generator**

1. **Problem Statement : Write a Python program that generates a hierarchical representation of a directory structure, including files and subdirectories, and optionally saves it to a text file or displays it in a graphical format.**

**.**

**Code:**

**import os**

**def generate\_tree(directory, prefix=''):**

**"""Generates the directory tree structure as a string."""**

**tree = []**

**contents = sorted(os.listdir(directory))**

**pointers = ['├── '] \* (len(contents) - 1) + ['└── ']**

**for pointer, name in zip(pointers, contents):**

**path = os.path.join(directory, name)**

**tree.append(f"{prefix}{pointer}{name}")**

**if os.path.isdir(path):**

**extension = '│ ' if pointer == '├── ' else ' '**

**tree.extend(generate\_tree(path, prefix + extension))**

**return tree**

**def save\_tree\_to\_file(tree, filename='directory\_structure.txt'):**

**"""Saves the directory tree structure to a text file."""**

**with open(filename, 'w') as f:**

**for line in tree:**

**f.write(f"{line}\n")**

**def print\_tree(tree):**

**"""Prints the directory tree structure to the console."""**

**for line in tree:**

**print(line)**

**def main():**

**directory = input("Enter the directory path: ")**

**if not os.path.isdir(directory):**

**print("Invalid directory path")**

**return**

**tree = generate\_tree(directory)**

**print("Directory structure:")**

**print\_tree(tree)**

**save\_option = input("Would you like to save the structure to a text file? (yes/no): ").lower()**

**if save\_option == 'yes':**

**filename = input("Enter the filename (with .txt extension): ")**

**save\_tree\_to\_file(tree, filename)**

**print(f"Directory structure saved to {filename}")**

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

**main()**

**Conclusion:**

**This script searches for files in the current directory and its subdirectories based on a user-provided search term. It uses the tree command with the -af flags to list all files and directories in a tree-like format, and grep to filter results that contain the search term. The filtered results are then redirected to out.txt.**