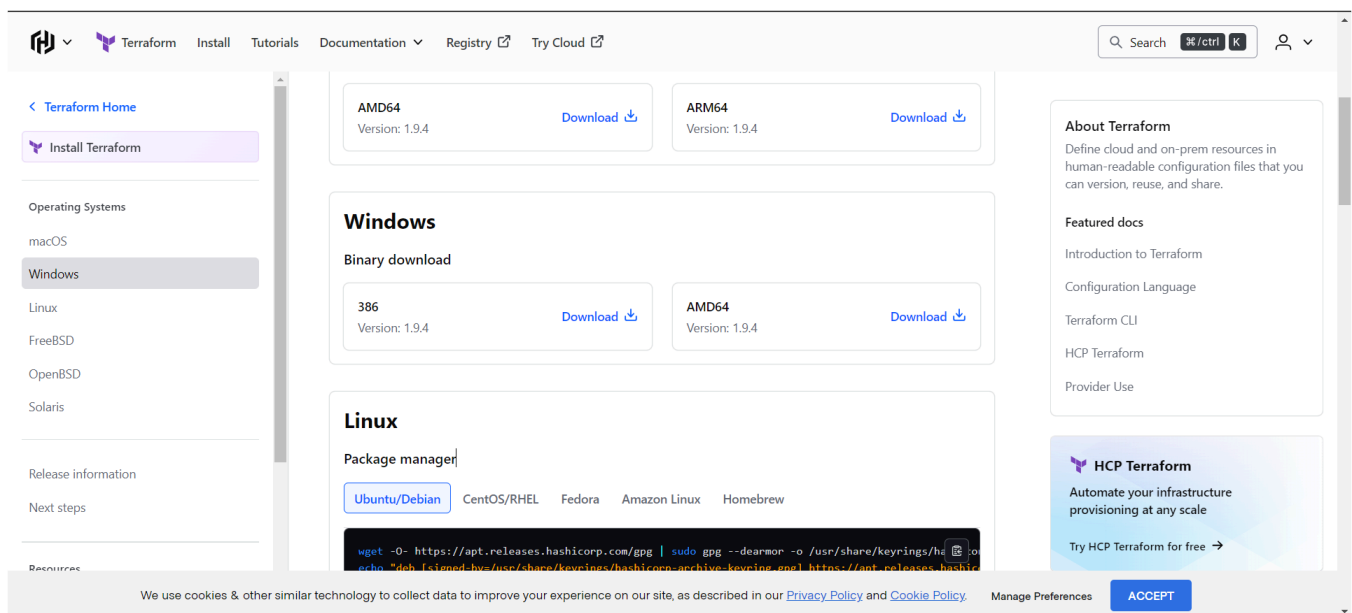**Aim:** To understand terraform lifecycle, and core concepts/terminologies, and install it on a Linux Machine or Windows.

Terraform: Terraform is an open-source Infrastructure as Code (IaC) tool developed by HashiCorp. It allows you to define, provision, and manage infrastructure across various cloud providers using a simple declarative configuration language. Terraform's key features include automation of infrastructure deployments, version control for infrastructure changes, and the ability to manage both cloud and on-premises resources in a consistent manner.

Step 1: Visit the website: https://developer.hashicorp.com/terraform/install?product_intent=terraform and install the terraform according to your system. Here we are downloading it for Windows.
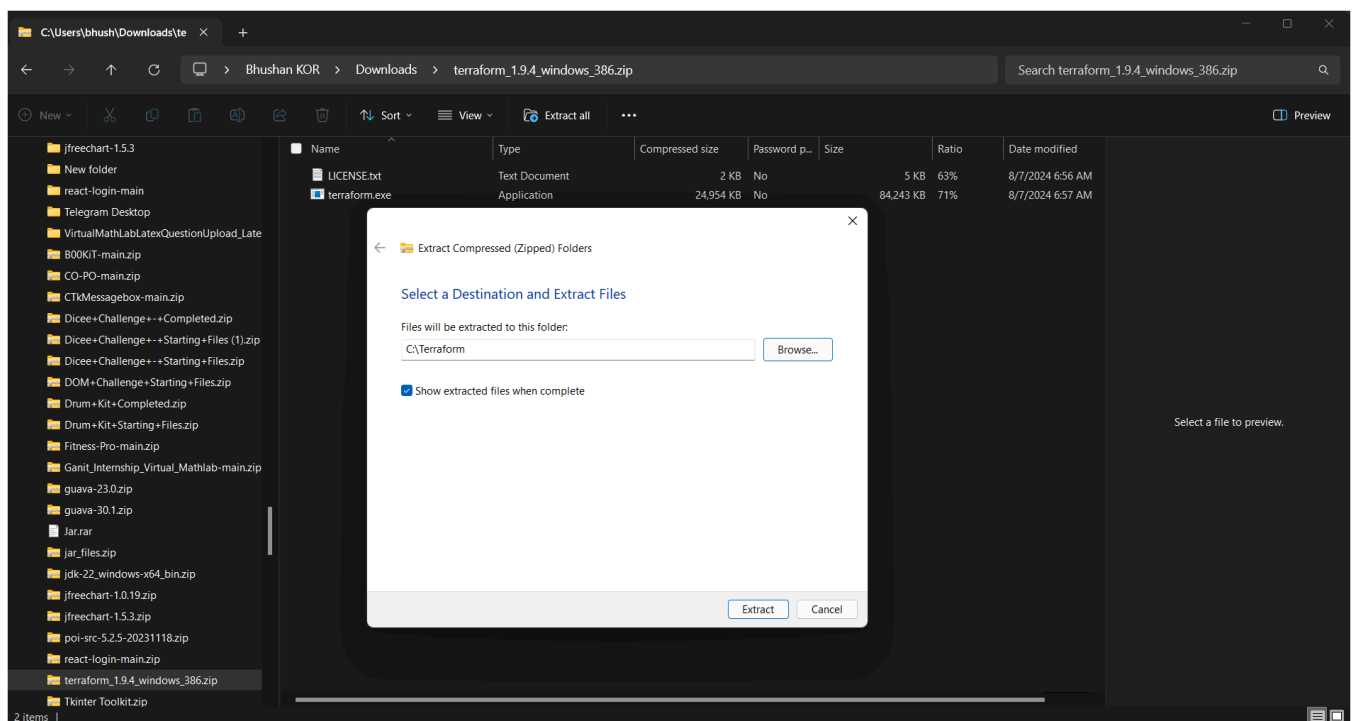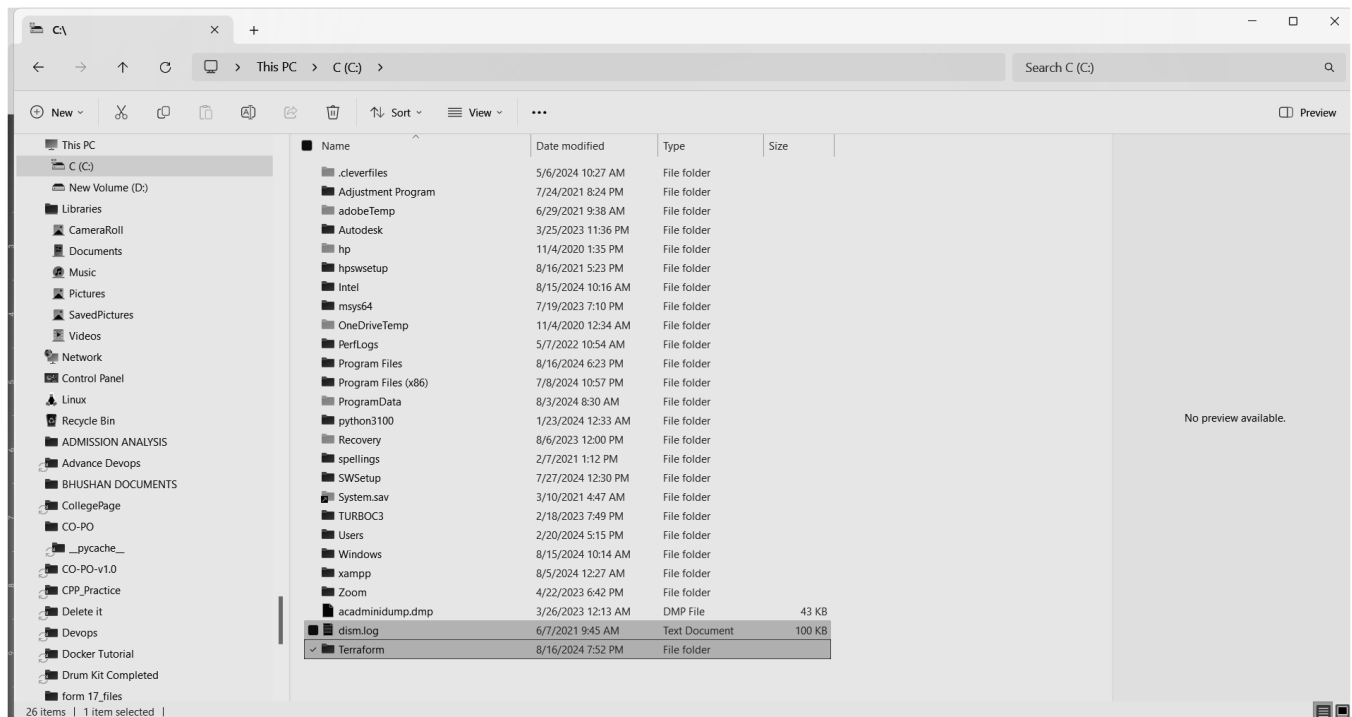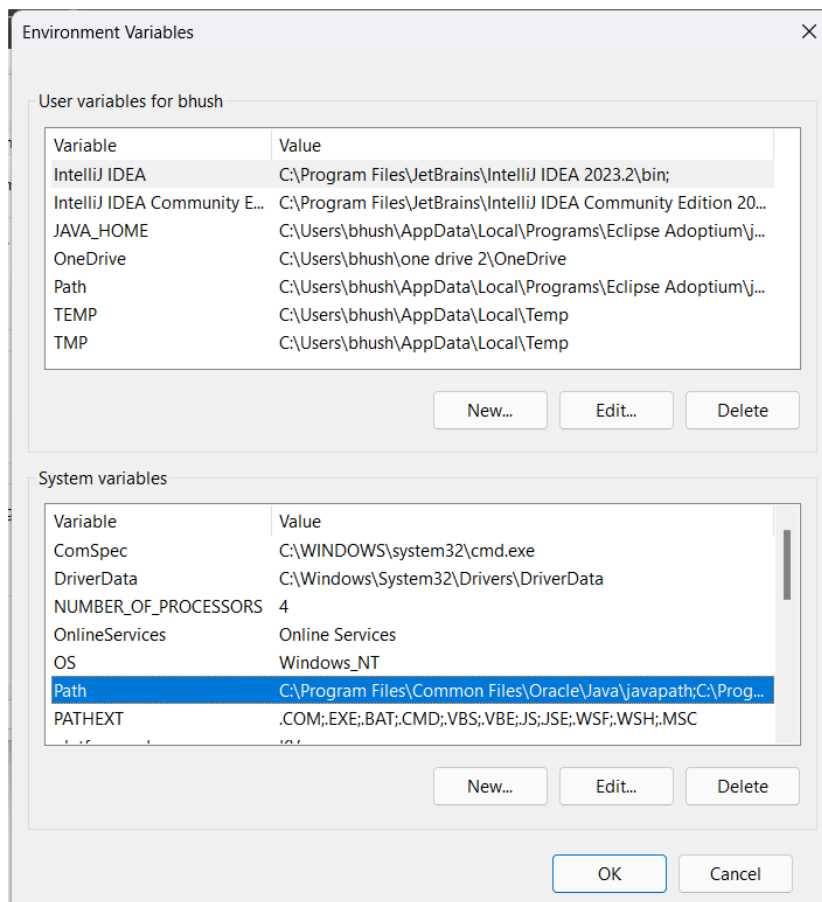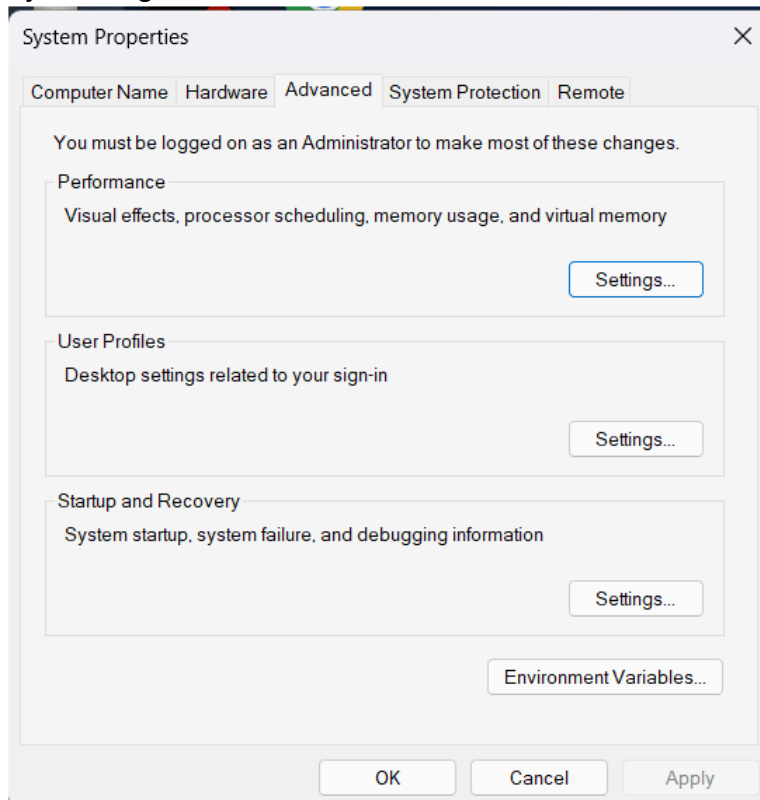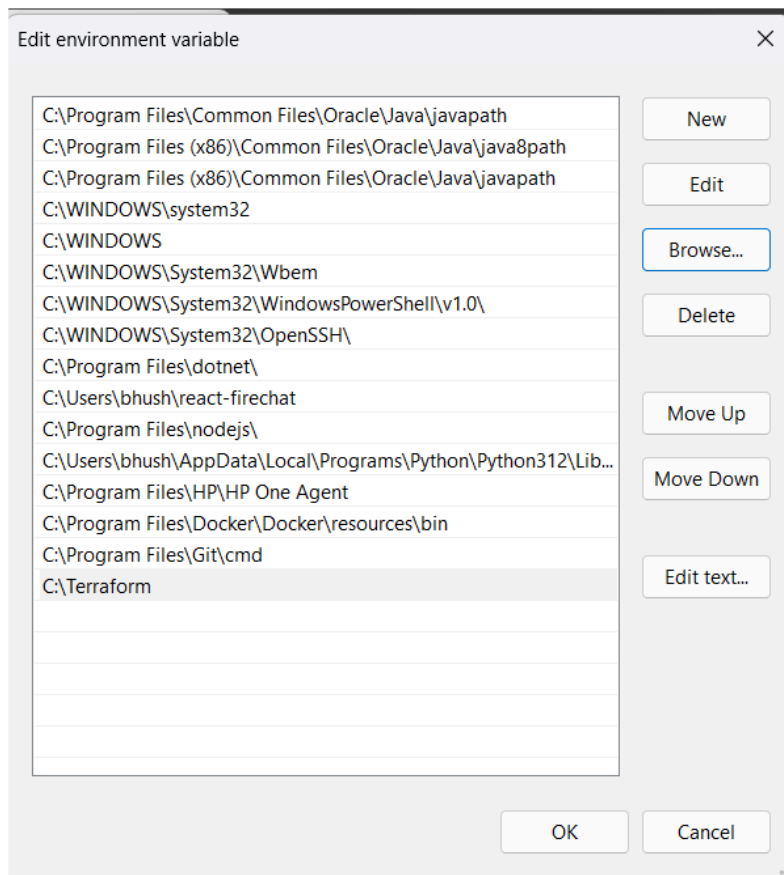
Step 2: Done Downloaded Successfully Now create the folder named Terraform in C: directory and extract the file in C:\Terraform.

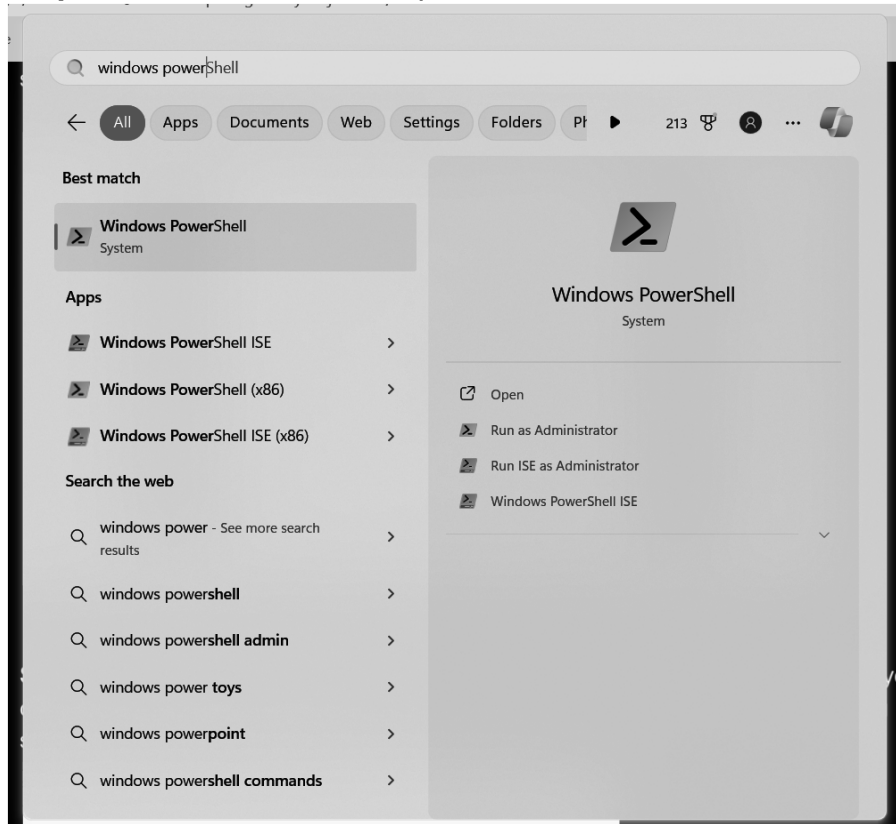**Step 3:** Edit the environment variable. In the path of system variables add the path of terraform by clicking edit. After that save and close it.

Step 4: Search for Windows power shell and Run it as administrator.

Step 5: Now run the command terraform in Windoes Powershell.

```
Administrator: Windows PowerShell (x86)
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  test          Execute integration tests for Terraform modules
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help         Show this help output, or the help for a specified subcommand.
  -version      An alias for the "version" subcommand.
PS C:\WINDOWS\system32>
```

Step 6: Congratulations You have successfully installed Terraform in your system.