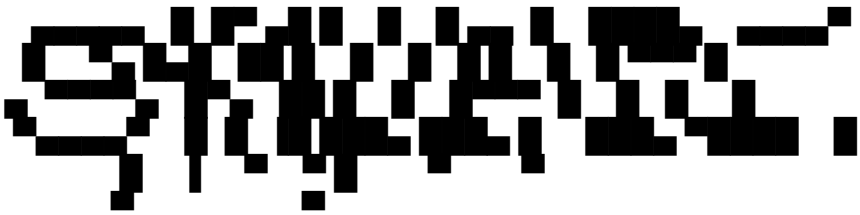
```
#!/bin/bash

## Copyright (c) 2023 mangalbhaskar. All Rights Reserved.
##__author__ = 'mangalbhaskar'
###------------------------------------------------------------
## Fingerprint Banner for Educational Screenshots
###------------------------------------------------------------


function lsd-mod.fio.yesno_no() {
  ## default is No
  local msg
  [[ $# -eq 0 ]] && msg="Are you sure" || msg="$*"
  msg=$(echo -e "\e[1;36m${msg}? \e[1;37m\e[1;31m[y/N]\e[1;33m>\e[0m ")
  [[ $(read -e -p "${msg}"; echo ${REPLY}) == [Yy]* ]] && return 0 || return -1
}


###------------------------------------------------------------

function lsd-mod.fingerprint.banner.skillplot() {

(>&2 echo -e "
###------------------------------------------------------------------------
```



```
>>> LSCRIPTS      : $(date +"%H:%M:%S, %d-%b-%Y, %A")
>>> Documentation : https://lscripts.skillplot.org
>>> Code          : https://github.com/skillplot/lscripts-docker
")
}


function lsd-mod.fingerprint.command_exists() {
  ## Function to check if a command is available
  command -v "$1" >/dev/null 2>&1
}

function lsd-mod.fingerprint.install_package_apt() {
  ## Function to install a package using apt (Debian/Ubuntu)
  sudo apt update -y
  sudo apt install -y "$1"
}

function lsd-mod.fingerprint.install_package_yum() {
  ## Function to install a package using yum (CentOS/RHEL)
  sudo yum install -y "$1"
}
```

```
# function lsd-mod.fingerprint.install_dependencies() {
#   ## Function to install required dependencies
#   # if ! lsd-mod.fingerprint.command_exists dmidecode; then
#   #   echo "dmidecode is not installed. Installing..."
#   #   lsd-mod.fingerprint.install_package_apt dmidecode || install_package_yum dmidecode
#   # fi

#   local _cmd="figlet"
#   type ${_cmd} &>/dev/null || {
#     (>&2 echo -e "figlet is not installed.")
#     (>&2 echo -e "Installing figlet... sudo access is required!")
#     lsd-mod.fingerprint.install_package_apt ${_cmd} || install_package_yum ${_cmd}
#   }
# }


function lsd-mod.fingerprint.get_mac_address() {
  ## Function to get the MAC address
  # ip link show | awk '/ether/ && !/link\/ether/ {print $2; exit}'
  # echo $(LANG=C ip link show | awk '/link\/ether/ {print $2}' | tr '\n' '|')
  echo $(ip link | awk '{print $2}'  | tr '\n' '|')
}

function lsd-mod.fingerprint.get_hostname() {
  ## Function to get the hostname
  hostname
}

function lsd-mod.fingerprint.get_cpu_info() {
  ## Function to get CPU information
  cat /proc/cpuinfo | grep 'model name' | head -n 1
}

function lsd-mod.fingerprint.get_os_name() {
  ## Function to obtain the operating system name (fallback if lsb_release is unavailable)
  if lsd-mod.fingerprint.command_exists lsb_release; then
    lsb_release -d | awk -F'\t' '{print $2}'
  else
    # Fallback method to obtain OS name
    if [ -e /etc/os-release ]; then
      grep -oP 'PRETTY_NAME="\K[^"]+' /etc/os-release
    elif [ -e /etc/lsb-release ]; then
      grep -oP 'DISTRIB_DESCRIPTION="\K[^"]+' /etc/lsb-release
    else
      echo "Unknown OS"
    fi
  fi
}

function lsd-mod.fingerprint.banner.system() {
  ## Function to generate a unique system ID by hashing the input
  local mac_address=$(lsd-mod.fingerprint.get_mac_address)
  local hostname=$(lsd-mod.fingerprint.get_hostname)
  local cpu_info=$(lsd-mod.fingerprint.get_cpu_info)
  local cpu_cores=$(grep -c '^processor' /proc/cpuinfo)
```

```
  local cpu_threads=$(grep -c '^processor' /proc/cpuinfo)
  local architecture=$(uname -m)
  local kernel_version=$(uname -r)
  local os_name=$(lsd-mod.fingerprint.get_os_name)

  local total_ram=$(free -h --si | awk '/Mem:/{print $2}')
  local storage_available=$(df -h / | awk '/\//{print $4}')

  local current_user=$(whoami)


  local combined_info="$mac_address-$hostname-$cpu_info-$cpu_cores-$cpu_threads-$architecture-$ke
rnel_version-$os_name-$total_ram-$current_user"

  ## Hash the combined system information
  local system_id=$(echo -n "$combined_info" | sha256sum | awk '{print $1}')


  # ## Try to get the system UUID from dmidecode if available
  # if lsd-mod.fingerprint.command_exists dmidecode; then
  #   vm_uuid=$(sudo dmidecode -s system-uuid 2>/dev/null)
  #   if [ -n "$vm_uuid" ]; then
  #     system_id="$system_id\nVM UUID: $vm_uuid"
  #   fi
  # fi

  (>&2 echo -e "System ID: $system_id
::$hostname
CPU Info: $cpu_info
CPU Cores: $cpu_cores
CPU Threads: $cpu_threads
Architecture: $architecture
Kernel Version: $kernel_version
OS Name: $os_name
Total RAM: $total_ram
Storage Available: $storage_available
Current User: $current_user ")
}


function lsd-mod.fingerprint.display_datetime() {
  ## Function to display the current date, time, month, and year
  local current_date=$(date "+%A, %B %d, %Y")
  local current_time=$(date "+%T")
  echo "Current Date: $current_date"
  echo "Current Time: $current_time"
}


## Function to check if the input is non-empty
function lsd-mod.fingerprint.is_non_empty() {
  [ -n "$1" ]
}

function lsd-mod.fingerprint.is_valid_email() {
```

```
  ## Function to validate the email format
  local email_regex="^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$"
  [[ "$1" =~ $email_regex ]]
}


function lsd-mod.fingerprint.get_valid_input() {
  ## Function to get a valid input (non-empty and matching the specified data type)
  local prompt="$1"
  local validation_func="$2"
  local input=""

  while true; do
    read -p "$prompt: " input
    if $(lsd-mod.fingerprint.is_non_empty "$input" && $validation_func "$input"); then
      break
    fi
  done

  echo "$input"
}


function lsd-mod.fingerprint.convert_to_output() {
  ## Function to convert input to ASCII art using figlet or simple text output
  local input="$1"
  local _art=$input

  if lsd-mod.fingerprint.command_exists figlet; then
    _art=$(figlet "$input")
  fi

  (>&2 echo -e "$_art")
}


function lsd-mod.fingerprint.main() {
  local LSCRIPTS=$( cd "$( dirname "${BASH_SOURCE[0]}")" && pwd )

  local _default=no
  local _cmd="figlet"
  type ${_cmd} &>/dev/null || {
    # (>&2 echo -e "figlet is not installed.")
    local _que="Insall dependencies to print the ASCII ART banner"
    lsd-mod.fio.yesno_${_default} "${_que}" && {
      echo "Installing..."
      (>&2 echo -e "Installing figlet... sudo access is required!")
      lsd-mod.fingerprint.install_package_apt ${_cmd} || install_package_yum ${_cmd}
    }
  }


  ## Get a valid name
  local user_name=$(lsd-mod.fingerprint.get_valid_input "Enter your name" "lsd-mod.fingerprint.is_non_e
mpty")
```

```
  ## Get a valid email
  local user_email=$(lsd-mod.fingerprint.get_valid_input "Enter your email" "lsd-mod.fingerprint.is_valid_e
mail")


  ## Get a valid student id
  local student_id=$(lsd-mod.fingerprint.get_valid_input "Enter your Student ID" "lsd-mod.fingerprint.is_no
n_empty")

  lsd-mod.fingerprint.banner.skillplot


  # ## Print ASCII art of the input or use simple text output
  lsd-mod.fingerprint.convert_to_output "$student_id"

  local combined_info=$(lsd-mod.fingerprint.banner.system)
  (>&2 echo -e "$combined_info")

  (>&2 echo -e "Thank you, $user_name! Details you provided are - ID: $student_id and Email: $user_em
ail.
###-------------------------------------------------------------------------")
}

lsd-mod.fingerprint.main "$@"
```