# 🔹 Auto-start Deskflow on Kali Linux (GUI)

**🔹 Auto-start Deskflow on Kali Linux (GUI)**

To make **Deskflow** start automatically with your desktop environment, create an **autostart entry**:

**1. Create the autostart directory (if it doesn’t exist)**

mkdir -p ~/.config/autostart

**2. Create a .desktop file for Deskflow**

nano ~/.config/autostart/deskflow.desktop

**3. Add the following content**

*(Adjust the Exec path if Deskflow is not in PATH)*

[Desktop Entry]

Type=Application

Exec=deskflow

Hidden=false

NoDisplay=false

X-GNOME-Autostart-enabled=true

Name=Deskflow

Comment=Start Deskflow automatically

**4. Save and exit**

Press **CTRL + O → ENTER → CTRL + X** to save and close Nano.

✅ Done! Now Deskflow will launch automatically every time you log into Kali’s GUI.

👉 Do you want me to also add a **section to verify if Deskflow is running after login**?

# Kali resolution

#kali resolution .sh

#!/bin/bash

# Detect HDMI screen (first connected)

HDMI=$(xrandr | grep -oP '^HDMI-\d+(?= connected)' | head -n1)

if [ -n "$HDMI" ]; then

# Add 1440x900 mode if it doesn't exist yet

if ! xrandr | grep -q "1440x900\_59.887"; then

xrandr --newmode "1440x900\_59.887" 106.50 1440 1528 1672 1904 900 903 909 934 -hsync +vsync

fi

# Add the mode to HDMI

xrandr --addmode "$HDMI" "1440x900\_59.887"

# Apply the mode

xrandr --output "$HDMI" --mode "1440x900\_59.887"

else

echo "⚠️ No HDMI screen detected."

fi

exit 0

# for dell inspillion-kali

Here’s your text in a **well-formatted way** with headings, code blocks, and proper spacing — **no changes to the content**:

**Dell Inspiron – Kali Linux Setup**

sudo apt update

sudo apt full-upgrade -y

sudo reboot

sudo apt install -y kali-desktop-xfce

sudo apt install -y lightdm

sudo dpkg-reconfigure lightdm

sudo systemctl start lightdm

# or

sudo reboot

**Change Terminal as Default**

**TTY mode in Kali Linux = text-only terminal without GUI.**

sudo apt update

sudo apt install -y kali-desktop-xfce lightdm

sudo systemctl set-default multi-user.target

sudo systemctl start lightdm

# or

startx

If you want the default as graphical:

sudo systemctl set-default graphical.target

**Switching Back to Terminal**

Ctrl + Alt + F1

sudo systemctl stop lightdm

**Check Wi-Fi Blockage**

rfkill list

Change for hardware switch: like turn off list.  
Many laptops have a Wi-Fi button (sometimes Fn + F2, F12, or a dedicated key).

**Network Scan**

sudo iwlist wlan0 scan

sudo iwlist wlan0 scan | grep ESSID

**Connect to Wi-Fi**

nmcli device status

sudo nmcli device wifi connect "MyWiFiNetwork" password "your\_password"

nmcli connection show

nmcli device show wlan0

**Test Connection**

ping google.com

# for kali

Here’s your text in a **well-formatted way** with headings, code blocks, and proper spacing — **no changes to the content**:

**Bluetooth Installation on Kali Linux**

sudo systemctl enable bluetooth

sudo systemctl start bluetooth

clear

sudo apt install blueman

**Install Google Chrome**

YouTube guide: <https://www.youtube.com/watch?v=uaxfiCjD9MM>

sudo apt install ./google-chrome-stable\_current\_amd64.deb

sudo apt install -f

google-chrome

**GitHub Setup**

**Token Name:**  
kcchero@1234

**Personal Access Token:**

github\_pat\_11A3D3Y2I0TwuYzE6fWVZb\_UNvAdv9Tet81t5UXQFS5xZw0wruvIL48ggyoBKckotbWTBJXB57wtAFDc27

**Commit from Visual Studio Code:**

Ctrl+Shift+G

**Configure Git globally:**

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

git config --global user.name "Salim Shrestha"

git config --global user.email "salim9shrestha@gmail.com"

**Check global config file:**

cat ~/.gitconfig

**Commit an empty change:**

git commit --allow-empty -m "test"

**Push changes:**

git push

This may ask for username if the remote URL uses HTTPS.

**Check remote URL:**

git remote -v

If the URL is like https://github.com/username/repo.git, your username is in the URL.

**Worked example:**

root㉿kali)-[/home/kali/Documents/journal-2]

└─# cat ~/.gitconfig

[user]

name = Salim Shrestha

email = salim9shrestha@gmail.com

git commit --allow-empty -m "test"

Ctrl+Shift+G

**Useful Commands**

shutdown now

**Vim Editor Commands**

:set number

:set nonumber

**Turn Off Display**

xset dpms force off

sleep 20; xset dpms force off

**Cheese Camera**

cheese

**Winsurf AI (Coding Tool)**

* Website: <https://windsurf.com/>
* Use for coding HTML, CSS, and task management apps.

**Time Tracker Tasks:**

1. TMA
2. Simple login & register page

* CRUD task
* Progress

**ChatGPT Prompt Guidance:**

* Need detailed prompt for Winsurf to build task management app using HTML, CSS, MySQL.
* Features: simple login & register, CRUD operations, progress dashboard, motivational quotes.

**Database Encryption:**

* Use **hash** for one-way encryption.

**Kali Web Directory**

(kali㉿kali)-[/var/www/html]

└─$ ls

index.html index.nginx-debian.html index.php

**Disk Formatting Commands**

lsblk

sudo umount /dev/sda1

sudo mkfs.ntfs -f /dev/sda1

sudo fdisk /dev/sda

sudo mkfs.ntfs -f /dev/sda1

lsblk -f

**Graphical Option**

sudo gparted

To format the entire disk with a graphical interface.

This keeps everything exactly as you wrote but **organized with headings, code blocks, and spacing** for readability.

# for xubuntu

Here’s your text in a **well-formatted way** with proper sections, code blocks, and headings — **no changes to the content**:

######################################

**Steps to Install Deskflow on Xubuntu**

✅ **What you can do instead on Xubuntu**

Here are the practical alternatives:

**Option A: Use Flatpak (best option)**

Deskflow is available on Flathub.  
[Flathub - Apps for Linux](https://flathub.org/)

**Install Flatpak (if not already):**

sudo apt update

sudo apt install flatpak

**Add the Flathub repository:**

sudo flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo

**Install Deskflow via Flatpak:**

flatpak install flathub org.deskflow.deskflow

**Run it:**

flatpak run org.deskflow.deskflow

This tends to work across different distros and avoids dependency issues.

**One-line command:**

sudo apt update && sudo apt install -y flatpak && sudo flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo && sudo flatpak install -y flathub org.deskflow.deskflow

Then run:

flatpak run org.deskflow.deskflow

######################################

**To Run Deskflow on Startup**

**Option 2: Manual Autostart File**

You can create a .desktop file in your autostart folder:

mkdir -p ~/.config/autostart

nano ~/.config/autostart/deskflow.desktop

**Paste this:**

[Desktop Entry]

Type=Application

Exec=flatpak run org.deskflow.deskflow

Hidden=false

NoDisplay=false

X-GNOME-Autostart-enabled=true

Name=Deskflow

Comment=Start Deskflow automatically

Save and exit (Ctrl+O, Ctrl+X).

######################################

**Task Manager**

sudo apt install htop

htop

######################################

**Power Management**

Use **TLP** (power management tool):

sudo apt install tlp

sudo tlp start

It reduces power/heat → less fan noise.

######################################

**DroidCam and OBS Setup**

1. **DroidCam via Flatpak:**

flatpak install flathub com.obsproject.Studio

flatpak install flathub com.obsproject.Studio.Plugin.DroidCam

flatpak override --user --device=all com.obsproject.Studio

1. **Download and Install DroidCam Client (Debian/Ubuntu):**

wget -O /tmp/droidcam\_client\_amd64.deb https://droidcam.app/go/droidCam.client.setup.deb

sudo apt install -f /tmp/droidcam\_client\_amd64.deb

**Video:**  
You will need the v4l2loopback module for video sharing with other apps. This allows you to create "virtual video devices".

sudo apt install linux-headers-$(uname -r) v4l2loopback-dkms

Needed an OBS to run, misunderstood.

**Method 2: Install OBS via Flatpak (latest version)**

**Install Flatpak (if not installed):**

sudo apt install -y flatpak

**Add the Flathub repository:**

flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo

**Install OBS Studio:**

flatpak install -y flathub com.obsproject.Studio

**Run OBS Studio:**

flatpak run com.obsproject.Studio

**Pros:** Always latest version, updates via Flatpak.  
**Cons:** Slightly larger install.

// But not able to loop.

######################

# Chat gpt on tty mode:

## ollama

**3. If you have *no GUI and no API* (pure TTY, offline)**

You can install a **local LLM** like **llama.cpp**, **ollama**, or **GPT4All** and run it in the terminal without internet.  
Example with Ollama:

curl -fsSL https://ollama.com/install.sh | sh

ollama run llama3

Now you can chat directly inside TTY.

#4.7bg .

Required high RAM 6gb minimum

## Chatgpt

Perfect 👍 Let’s set up **Shell-GPT (sgpt)** on Kali so you can have **real ChatGPT inside your TTY**.

**🔹 Step 1. Make sure Python & pip are installed**

Run:

python3 --version

pip3 --version

If pip3 is missing:

sudo apt update

sudo apt install python3-pip -y

**🔹 Step 2. Install Shell-GPT**

Run:

pip3 install shell-gpt

#this one working.

sudo apt install pipx

pipx install shell-gpt

**🔹 Step 3. Get your OpenAI API key**

1. Go to 👉 [OpenAI API Keys](https://platform.openai.com/account/api-keys)
2. Create a new **secret key**.
3. Copy it.

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

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

**🔹 Step 4. Set your API key in terminal**

Run this (replace with your actual key):

export OPENAI\_API\_KEY="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"

👉 To make it permanent (so you don’t set it every time), add the line to your ~/.bashrc or ~/.zshrc:

echo 'export OPENAI\_API\_KEY="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"' >> ~/.bashrc

source ~/.bashrc

**🔹 Step 5. Use ChatGPT in TTY 🎉**

Examples:

* **Single question**

sgpt "Explain what TTY mode is in Kali Linux"

* **Chat mode**

sgpt --chat mysession

Now it behaves like a **full ChatGPT chat**. You type → it replies → continues history.

* **Multiple sessions**

sgpt --chat linuxhelp

sgpt --chat funchat

**🔹 Step 6. Optional – Make it fancy**

* To clear old chats:

sgpt --chat mysession --clear

* To use a specific model (like GPT-4o-mini):

sgpt --model gpt-4o-mini "summarize this text"

✅ That’s it! Now you can have **real ChatGPT inside pure TTY mode**.

Do you want me to also show you how to make it look like a **perfect chat interface** (with your text as You: and ChatGPT as AI:), instead of plain outputs?