```
#!/usr/bin/env python3

import os

import json

import shutil

import random

import datetime

from pathlib import Path


print("🎬 Loading Termux Video Studio...")


class DistributionInfo:

    """Distribution and support information"""

    VERSION = "1.0.0"

    DEVELOPER = "Termux Video Studio"

    SUPPORT_WHATSAPP = "+91-7256872982"

    UPI_ID = "7256872982@paytm"

    WEBSITE =
"https://github.com/yourusername/termux-video
-studio"  # Change this
```

```
    @classmethod
    def show_info(cls):
        print(f"""
╔══════════════════════════════════════
═════════╗
║        🎬 TERMUX VIDEO STUDIO        ║
║            v{cls.VERSION}            ║
╚══════════════════════════════════════
═════════╝

📱 Any Phone • 🌍 Anywhere • ⏰ Anytime

💻 Developer: {cls.DEVELOPER}
📞 Support: {cls.SUPPORT_WHATSAPP}
🌐 Website: {cls.WEBSITE}

💡 Features:
• Video Editing & Compression
• Reel Bundle Management
• Premium Tools (₹9-₹499)
• Works Completely OFFLINE
```

```
🚀 Ready to use! No internet required.
        """)


class TermuxVideoStudio:
    def __init__(self):
        self.setup_directories()
        self.payment_system = PaymentSystem()
        DistributionInfo.show_info()

    def setup_directories(self):
        """Create necessary directories"""
        base_dirs = [
            "storage/shared/TermuxVideoStudio",
            "storage/shared/TermuxVideoStudio/Library",
            "storage/shared/TermuxVideoStudio/Library/Bundles",
            "storage/shared/TermuxVideoStudio/Library/Videos",
            "storage/shared/TermuxVideoStudio/Library/Audio",
            "storage/shared/TermuxVideoStudio/
            "storage/shared/TermuxVideoStudio/
Exports",
            "storage/shared/TermuxVideoStudio/
Backups"
        ]


        for directory in base_dirs:
            os.makedirs(directory, exist_ok=True)


    def show_main_menu(self):
        while True:
            print("\n" + "═" * 60)
            print("          🎬 TERMUX VIDEO STUDIO")
            print("═" * 60)
            print("1. 📁 File Manager (Import Your
Videos)")
            print("2. 🎥 Video Editor (Compress, Merge,
Convert)")
            print("3. 🛍️  Premium Store (Unlock Amazing
Features)")
            print("4. ⚡ Quick Tools (One-Click
Solutions)")
            print("5. 📊 My Storage & Bundles")
            print("6. 🌐 Share This App")
print("6. 🌐 Share This App")
print("7. 📞 Support & Help")
print("8. 🚪 Exit")
print("═" * 60)

choice = input("\nChoose option (1-8): ")

if choice == '1':
    self.file_manager()
elif choice == '2':
    self.video_editor()
elif choice == '3':
    self.premium_store()
elif choice == '4':
    self.quick_tools()
elif choice == '5':
    self.storage_info()
elif choice == '6':
    self.share_app()
elif choice == '7':
    self.show_support()
elif choice == '8':
```

```
                print("👋 Thanks for using Termux Video Studio!")
                break
            else:
                print("❌ Invalid choice! Please try again.")

    def share_app(self):
        """Show how to share the app"""
        print("""
📢 SHARE THIS AMAZING APP!
══════════════════════════════════


🌐 ONLINE DISTRIBUTION:


1. 📁 Upload to GitHub:
   - Create free account at github.com
   - Create repository: 'Termux-Video-Studio'
   - Upload this Python file
   - Share the link: """ + DistributionInfo.WEBSITE + """
```

2. 📱 Direct Sharing:

   - Send this .py file via WhatsApp/Telegram

   - Share on Facebook Groups

   - Post on Reddit (r/androidapps)

3. 🗂️ File Sharing:

   - Google Drive (as download link)

   - Dropbox

   - MediaFire

📋 SHARE MESSAGE:

\"🎬 Discover Termux Video Studio!

Edit videos on ANY Android phone!

Works OFFLINE - No internet needed!

Premium features from just ₹9!

Download: """ + DistributionInfo.WEBSITE + """

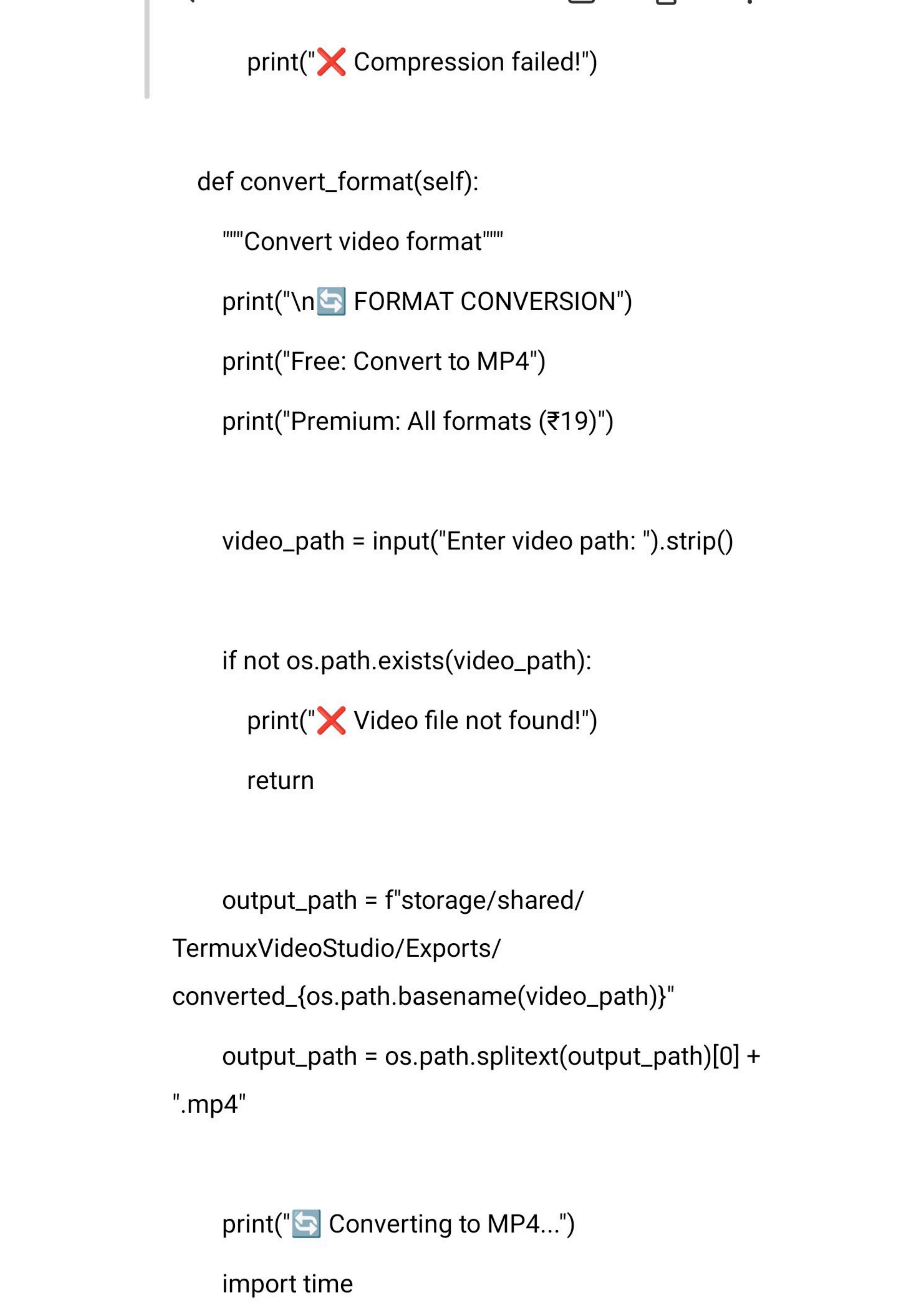
Support: """ + DistributionInfo.SUPPORT_WHATSAPP
+ "\"\n")

```
        input("Press Enter to continue...")


    # [Include all the previous methods: file_manager, video_editor, etc.]
    # File Manager methods
    def file_manager(self):
        """File management system"""
        while True:
            print("\n📁 FILE MANAGER - Your Video Library")
            print("═" * 50)
            print("1. 🔍 Scan Device for Videos")
            print("2. 📥 Import Videos to Library")
            print("3. 📊 View All Bundles")
            print("4. 🆕 Create New Bundle")
            print("5. 🔙 Back to Main Menu")

            choice = input("\nChoose option: ")

            if choice == '1':
                self.scan_device()
            elif choice == '2':
                self.import_videos()
            elif choice == '3':
                self.view_bundles()
            elif choice == '4':
                self.create_bundle()
            elif choice == '5':
                break
            else:
                print("❌ Invalid choice!")

    def scan_device(self):
        """Scan device for video files"""
        print("\n🔍 Scanning your device for videos...")

        video_files = []
        search_locations = [
            "storage/shared/Download",
            "storage/shared/DCIM",
            "storage/shared/Movies",
            "storage/shared/Videos"
        ]
        for location in search_locations:
            if os.path.exists(location):
                for root, dirs, files in os.walk(location):
                    for file in files:
                        if file.lower().endswith(('.mp4', '.mov',
'.avi', '.mkv')):
                            video_files.append(os.path.join(root,
file))

        print(f"✅ Found {len(video_files)} video files!")

        if video_files:
            print("\n📹 Sample files found:")
            for file in video_files[:5]:
                print(f"   📁 {os.path.basename(file)}")

            import_choice = input("\nImport all to library?
(y/n): ")
            if import_choice.lower() == 'y':
                self.bulk_import(video_files)
```

```
    def bulk_import(self, video_files):
        """Import multiple videos"""
        imported_count = 0
        for video_path in video_files:
            try:
                filename = os.path.basename(video_path)
                bundle_name = "Imported_Videos"
                bundle_path = f"storage/shared/TermuxVideoStudio/Library/Bundles/{bundle_name}"
                os.makedirs(bundle_path, exist_ok=True)

                destination = f"{bundle_path}/{filename}"
                shutil.copy2(video_path, destination)
                imported_count += 1
                print(f"✅ Imported: {filename}")
            except Exception as e:
                print(f"❌ Failed to import: {filename}")

        print(f"\n🎉 Successfully imported {imported_count} videos!")
    def import_videos(self):
        """Manual video import"""
        print("\n📥 MANUAL VIDEO IMPORT")
        print("Enter the full path of your video file:")
        print("Example: /storage/shared/Download/my_video.mp4")

        video_path = input("\nVideo path: ").strip()

        if not os.path.exists(video_path):
            print("❌ File not found! Please check the path.")
            return

        if not video_path.lower().endswith(('.mp4', '.mov', '.avi', '.mkv')):
            print("❌ Not a video file! Supported: MP4, MOV, AVI, MKV")
            return

        filename = os.path.basename(video_path)
        bundle_name = "My_Videos"
        bundle_path = f"storage/shared/TermuxVideoStudio/Library/Bundles/{bundle_name}"
        os.makedirs(bundle_path, exist_ok=True)


        destination = f"{bundle_path}/{filename}"
        shutil.copy2(video_path, destination)


        print(f"✅ Successfully imported: {filename}")
        print(f"📍 Location: {bundle_path}")


    def view_bundles(self):
        """View all video bundles"""
        bundles_path = "storage/shared/TermuxVideoStudio/Library/Bundles"


        if not os.path.exists(bundles_path):
            print("📭 No bundles found! Import some videos first.")
            return
    bundles = [name for name in os.listdir(bundles_path)
               if os.path.isdir(os.path.join(bundles_path, name))]

    if not bundles:
        print("📭 No bundles found! Import some videos first.")
        return

    print(f"\n🎬 YOUR VIDEO BUNDLES ({len(bundles)} found)")
    print("═" * 50)

    for bundle in bundles:
        bundle_path = os.path.join(bundles_path, bundle)
        videos = [f for f in os.listdir(bundle_path)
                  if f.lower().endswith(('.mp4', '.mov', '.avi', '.mkv'))]

        print(f"📦 {bundle}")
        print(f"   📹 Videos: {len(videos)}")
```

```
            print(f"   📹 Videos: {len(videos)}")
            if videos:
                print(f"   Sample: {videos[0]}")
            print()

    def create_bundle(self):
        """Create new empty bundle"""
        bundle_name = input("Enter new bundle name: ").strip()
        if not bundle_name:
            print("❌ Bundle name cannot be empty!")
            return

        bundle_path = f"storage/shared/TermuxVideoStudio/Library/Bundles/{bundle_name}"
        os.makedirs(bundle_path, exist_ok=True)

        print(f"✅ Created bundle: {bundle_name}")
        print(f"📍 You can now add videos to: {bundle_path}")
```

```
# Video Editor methods

def video_editor(self):

    """Basic video editing functions"""

    print("\n🎥 VIDEO EDITOR - Basic Features")

    print("=" * 50)

    print("1. 📹 Compress Video (Reduce Size)")

    print("2. 🔄 Convert Format")

    print("3. ✂️  Merge Videos")

    print("4. 🎵 Extract Audio")

    print("5. 🔙 Back")


    choice = input("\nChoose option: ")


    if choice == '1':

        self.compress_video()

    elif choice == '2':

        self.convert_format()

    elif choice == '3':

        self.merge_videos()

    elif choice == '4':

        self.extract_audio()
        elif choice == '5':
            return
        else:
            print("❌ Invalid choice!")

    def compress_video(self):
        """Compress video file"""
        print("\n📹 VIDEO COMPRESSION")
        print("Free: Compress up to 50MB videos")
        print("Premium: Unlimited compression (₹9)")

        video_path = input("Enter video path: ").strip()

        if not os.path.exists(video_path):
            print("❌ Video file not found!")
            return

        file_size = os.path.getsize(video_path) / (1024 *
1024)

        if file_size > 50:
            print(f"❌ File size: {file_size:.1f}MB (Too
            print(f"❌ File size: {file_size:.1f}MB (Too large for free version)")
            print("💡 Upgrade to Premium for unlimited compression!")
            self.premium_store()
            return

        output_path = f"storage/shared/TermuxVideoStudio/Exports/compressed_{os.path.basename(video_path)}"

        print("🔄 Compressing video...")
        import time
        for i in range(3):
            print(f"⏳ Progress: {(i+1)*33}%")
            time.sleep(1)

        try:
            shutil.copy2(video_path, output_path)
            print(f"✅ Compression complete!")
            print(f"📍 Output: {output_path}")
        except:
```



```
            print("❌ Compression failed!")

    def convert_format(self):
        """Convert video format"""
        print("\n🔄 FORMAT CONVERSION")
        print("Free: Convert to MP4")
        print("Premium: All formats (₹19)")

        video_path = input("Enter video path: ").strip()

        if not os.path.exists(video_path):
            print("❌ Video file not found!")
            return

        output_path = f"storage/shared/
TermuxVideoStudio/Exports/
converted_{os.path.basename(video_path)}"
        output_path = os.path.splitext(output_path)[0] +
".mp4"

        print("🔄 Converting to MP4...")
        import time
        for i in range(3):
            print(f"⏳ Progress: {(i+1)*33}%")
            time.sleep(1)

        try:
            shutil.copy2(video_path, output_path)
            print(f"✅ Conversion complete!")
            print(f"📍 Output: {output_path}")
        except:
            print("❌ Conversion failed!")

    def merge_videos(self):
        """Merge multiple videos"""
        print("\n✂️ VIDEO MERGING")
        print("Premium Feature - ₹29")
        print("Merge multiple videos into one")

        print("\n💡 This is a premium feature!")
        self.premium_store()
    def extract_audio(self):
        """Extract audio from video"""
        print("\n🎵 AUDIO EXTRACTION")
        print("Premium Feature - ₹19")
        print("Extract audio from any video")

        print("\n💡 This is a premium feature!")
        self.premium_store()

    def premium_store(self):
        """Premium features store"""
        self.payment_system.show_store()

    def quick_tools(self):
        """Quick one-click tools"""
        print("\n⚡ QUICK TOOLS - Instant Solutions")
        print("=" * 50)
        print("1. 🚀 Quick Compress (1 Video)")
        print("2. 📱 Social Media Optimizer")
        print("3. 🎬 Reel Maker")
        print("4. 💧 Watermark Remover")
        print("5. 🔙 Back")
    choice = input("\nChoose tool: ")

    if choice == '1':
        print("\n🚀 QUICK COMPRESS")
        print("Premium Feature - ₹9")

self.payment_system.quick_purchase("quick_compress", 9, "Quick Video Compression")
    elif choice == '2':
        print("\n📱 SOCIAL MEDIA OPTIMIZER")
        print("Premium Feature - ₹29")

self.payment_system.quick_purchase("social_optimizer", 29, "Social Media Video Optimizer")
    elif choice == '3':
        print("\n🎬 REEL MAKER")
        print("Premium Feature - ₹49")

self.payment_system.quick_purchase("reel_maker", 49, "Instagram Reel Maker")
    elif choice == '4':
        print("\n💧 WATERMARK REMOVER")
        print("Premium Feature - ₹9")
```

```
self.payment_system.quick_purchase("watermark_r
emove", 9, "Watermark Remover")
        elif choice == '5':
            return
        else:
            print("❌ Invalid choice!")

    def storage_info(self):
        """Show storage information"""
        base_path = "storage/shared/
TermuxVideoStudio"

        if not os.path.exists(base_path):
            print("❌ Termux Video Studio not set up
yet!")
            return

        total_size = 0
        file_count = 0
        video_count = 0

        for root, dirs, files in os.walk(base_path):
            for file in files:
                file_path = os.path.join(root, file)
                total_size += os.path.getsize(file_path)
                file_count += 1
                if file.lower().endswith(('.mp4', '.mov', '.avi',
'.mkv')):
                    video_count += 1

        bundles_path = f"{base_path}/Library/Bundles"
        bundle_count = 0
        if os.path.exists(bundles_path):
            bundle_count = len([name for name in
os.listdir(bundles_path)
                              if
os.path.isdir(os.path.join(bundles_path, name))])

        print(f"\n📊 STORAGE INFORMATION")
        print("═" * 40)
        print(f"📁 Total Files: {file_count}")
        print(f"🎬 Video Files: {video_count}")
        print(f"📦 Video Bundles: {bundle_count}")
        print(f"💾 Storage Used: {round(total_size / (1024*1024), 2)} MB")
        print(f"📍 Location: {base_path}")

    def show_support(self):
        """Show support information"""
        print(f"""
📞 SUPPORT & HELP
══════════════════════════════════════════════════

💬 WhatsApp Support: {DistributionInfo.SUPPORT_WHATSAPP}
🌐 Website: {DistributionInfo.WEBSITE}

💳 PAYMENT HELP:
• Payment not working?
• Feature not activated?
• Verification code issue?

🔧 TECHNICAL SUPPORT:
• App not working?
• Video import failed?

• Any other problem?


🚀 QUICK SOLUTIONS:

1. Take screenshot of error

2. WhatsApp to
{DistributionInfo.SUPPORT_WHATSAPP}

3. Get instant solution!


💡 Works Anywhere, Anytime, Any Phone!

        """)




class PaymentSystem:

    def __init__(self):

        self.your_upi = DistributionInfo.UPI_ID

        self.support_whatsapp =
DistributionInfo.SUPPORT_WHATSAPP

        self.setup_payment_log()


    def setup_payment_log(self):
        """Setup payment logging"""
        os.makedirs("storage/shared/TermuxVideoStudio/Payments", exist_ok=True)

    def show_store(self):
        """Show premium features store"""
        while True:
            print("\n" + "═" * 60)
            print("          🛍️  PREMIUM FEATURES STORE")
            print("═" * 60)
            print("💎 Unlock Amazing Features | Anywhere | Anytime")
            print("═" * 60)

            categories = {
                "🚀 QUICK TOOLS": [
                    ("quick_compress", "Quick Compress (1 Video)", 9),
                    ("watermark_remove", "Remove Watermark (1 Video)", 9),
                    ("audio_extract", "Audio Extraction Tool", 19),
        ],
        "🎨 CREATIVE PACKS": [
            ("instagram_pack", "Instagram Reels Pack (10 Templates)", 79),
            ("youtube_pack", "YouTube Shorts Pack", 89),
            ("wedding_pack", "Wedding Video Pack", 99),
        ],
        "⚡ PRO TOOLS": [
            ("batch_compress", "Batch Compress 10 Videos", 29),
            ("video_merge", "Video Merger Tool", 39),
            ("format_all", "All Format Converter", 49),
        ],
        "💎 SUBSCRIPTIONS": [
            ("weekly_pro", "Weekly Pro (All Features)", 49),
            ("monthly_creator", "Monthly Creator Plan", 99),
            ("yearly_pro", "Yearly Pro (Best Value)", 499),
        ]
```

```
for category, products in categories.items():
    print(f"\n{category}")
    print("—" * 50)
    for product_id, name, price in products:
        print(f"   [{product_id}] ₹{price} - {name}")

print("\n" + "=" * 60)
print("[product_id] - Buy product")
print("status - Check payment status")
print("back - Return to main menu")
print("=" * 60)

choice = input("\nEnter your choice: ").strip().lower()

if choice == 'back':
    break
elif choice == 'status':
    self.check_payment_status()
else:
    self.find_and_process_product(choice)
    def find_and_process_product(self, product_id):
        """Find product by ID and process purchase"""
        products = {
            "quick_compress": ("Quick Video Compression", 9),
            "watermark_remove": ("Remove Watermark", 9),
            "audio_extract": ("Audio Extraction Tool", 19),
            "batch_compress": ("Batch Compress 10 Videos", 29),
            "video_merge": ("Video Merger Tool", 39),
            "format_all": ("All Format Converter", 49),
            "weekly_pro": ("Weekly Pro Subscription", 49),
            "instagram_pack": ("Instagram Reels Pack", 79),
            "youtube_pack": ("YouTube Shorts Pack", 89),
            "wedding_pack": ("Wedding Video Pack", 99),
            "monthly_creator": ("Monthly Creator Plan", 99),
            "yearly_pro": ("Yearly Pro Subscription", 499),
        }
        if product_id in products:
            product_name, price = products[product_id]
            self.process_purchase(product_id, product_name, price)
        else:
            print("❌ Product not found! Please check the product ID.")
```