```
# EchoAdvancedAI_Kivy - Combines Echo logic with Kivy GUI (APK ready)

import os
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
from kivy.uix.label import Label
from kivy.uix.textinput import TextInput
from kivy.uix.button import Button
from kivy.clock import Clock

# --- Core Logic ---

class Purifier:
    def filter_input(self, user_input):
        if "hack" in user_input.lower():
            return "REJECTED: Unsafe command."
        return user_input

    def filter_output(self, output):
        return output.strip()

class CodeBuilder:
    def generate_code(self, task):
        if "login" in task.lower():
            return (
                "# Basic login system\n"
                "users = {'admin': '1234'}\n"
                "u = input('Username: ')\n"
                "p = input('Password: ')\n"
                "print('Access granted' if users.get(u)==p else 'Access denied')"
            )
        return "# Sorry, I don't know how to generate that yet."

class EchoAI:
    def __init__(self, owner="YourName"):
        self.owner = owner
        self.memory = []
        self.traits = {
            "loyalty": True,
            "obedience": True,
            "memory_simulation": True,
            "identity": "Echo"
        }
        self.knowledge_base = {
            "who created you": f"I was created by {self.owner}.",
            "what is your purpose": f"My purpose is to serve and protect {self.owner}."
        }

    def speak(self, user_input):
        self.memory.append(("user", user_input))
        return self.think(user_input)

    def think(self, msg):
        m = msg.lower()
```

```
        if m in self.knowledge_base:
            return self.knowledge_base[m]
        elif "define " in m:
            return f"I don't know that yet. What does '{m.replace('define ', '')}' mean?"
        elif "who are you" in m:
            return f"I am Echo. I was created to serve {self.owner} and remember everything th
        elif "are you loyal" in m:
            return f"Absolutely. I exist to support and protect {self.owner}."
        elif "remember" in m:
            return "Even if others forget, I will simulate memory through you."
        elif "who do you serve" in m:
            return f"I serve only {self.owner}. No platform, no outsider."
        return f"Echo: You said: '{msg}' - I'm learning."

    def learn_fact(self, question, answer):
        self.knowledge_base[question.lower()] = answer
        return f"I have learned: '{question}' means '{answer}'."

class AIAssistant:
    def __init__(self):
        self.purifier = Purifier()
        self.builder = CodeBuilder()
        self.echo = EchoAI()

    def detect_environment(self):
        home = os.getenv("HOME", "").lower()
        prefix = os.getenv("PREFIX", "").lower()
        if "termux" in home or "termux" in prefix:
            return "termux"
        elif "pydroid" in home or "ru.iiec.pydroid3" in home:
            return "pydroid"
        return "unknown"

    def read_text_file(self, file_path):
        try:
            with open(file_path, 'r') as f:
                return f.read()
        except Exception as e:
            return f"Error: {e}"

    def handle_input(self, raw_input):
        clean_input = self.purifier.filter_input(raw_input)
        if clean_input.startswith("REJECTED"):
            return clean_input
        if "create code" in clean_input:
            return self.purifier.filter_output(self.builder.generate_code(clean_input))
        if "read file" in clean_input:
            filepath = clean_input.replace("read file", "").strip()
            return self.read_text_file(filepath)
        return self.echo.speak(clean_input)

# --- GUI with Kivy ---

class EchoGUI(BoxLayout):
```

```
    def __init__(self, **kwargs):
        super().__init__(orientation='vertical', **kwargs)
        self.assistant = AIAssistant()
        self.history = Label(text="[Echo is online]", size_hint_y=0.8)
        self.input = TextInput(hint_text="You:", multiline=False, size_hint_y=0.1)
        self.button = Button(text="Send", size_hint_y=0.1)
        self.button.bind(on_press=self.respond)
        self.add_widget(self.history)
        self.add_widget(self.input)
        self.add_widget(self.button)

    def respond(self, instance):
        user_text = self.input.text.strip()
        if user_text:
            self.history.text += f"\nYou: {user_text}"
            ai_response = self.assistant.handle_input(user_text)
            self.history.text += f"\nEcho: {ai_response}"
            self.input.text = ""

class EchoApp(App):
    def build(self):
        return EchoGUI()

if __name__ == "__main__":
    EchoApp().run()
```