```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 12,
      "metadata": {
        "id": "1Me1YCmEyDCG"
      },
      "outputs": [],
      "source": [
        "!pip install streamlit pyngrok --quiet"
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "!pkill -f ngrok"
      ],
      "metadata": {
        "id": "xiWWBEFy7wlj"
      },
      "execution_count": 21,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "!ngrok authtoken 2yD7JZWRLeAgDLmOWlY3Vi9b0zv_6NFgPaLQxKbNsCcuJTTmV"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "xTA94M_g2ZoK",
        "outputId": "cb72db9c-1c03-43c4-c5ae-0cbdb0e2b923"
      },
      "execution_count": 22,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
```

```
        "text": [
          "Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "%%writefile \"Smart_app.py\"\n",
      "import streamlit as st\n",
      "from transformers import pipeline, AutoTokenizer, AutoModelForCausalLM\n",
      "\n",
      "@st.cache_resource\n",
      "def load_model():\n",
      "    tokenizer = AutoTokenizer.from_pretrained(\"ibm-granite/granite-3.3-2b-instruct\")\n",
      "    model = AutoModelForCausalLM.from_pretrained(\"ibm-granite/granite-3.3-2b-instruct\")\n",
      "    instruct_pipeline = pipeline(\"text-generation\", model=model, tokenizer=tokenizer)\n",
      "    return instruct_pipeline\n",
      "\n",
      "model = load_model()\n",
      "\n",
      "st.title(\"SmartSDLC - AI-enhanced Software Development Life Cycle\")\n",
      "st.write(\"  This app is running from Google Colab using Streamlit + ngrok!\")\n",
      "\n",
      "menu = [\"Requirement Analysis\", \"Code Generation\", \"Code Review\", \"Test Case
Generation\"]\n",
      "choice = st.sidebar.selectbox(\"Select Stage\", menu)\n",
      "\n",
      "def generate_response(prompt, max_tokens=200):\n",
      "    output = model(prompt, max_new_tokens=max_tokens,
do_sample=False)[0][’generated_text’]\n",
      "    return output.replace(prompt, \"\").strip()\n",
      "\n",
      "if choice == \"Requirement Analysis\":\n",
      "    st.header(\"Requirement Analysis & Summarization\")\n",
      "    req_text = st.text_area(\"Paste your software requirements here:\")\n",
      "    if st.button(\"Summarize Requirements\"):\n",
      "        if req_text.strip():\n",
      "            prompt = f\"Summarize the following software
requirement:\\n\\n{req_text}\\n\\nSummary:\"\n",
      "            summary = generate_response(prompt, max_tokens=100)\n",
      "            st.success(\"Summary:\")\n",
      "            st.write(summary)\n",
      "        else:\n",
      "            st.warning(\"Please input requirements text.\")\n",
      "\n",
      "elif choice == \"Code Generation\":\n",
      "    st.header(\"Generate Code from Requirements\")\n",
      "    req_text = st.text_area(\"Describe the functionality you want to implement:\")\n",
      "    if st.button(\"Generate Code\"):\n",
      "        if req_text.strip():\n",
      "            prompt = f\"Generate Python code for the following functionality:\\n\\n{req_text}\\n\\nPython
code:\"\n",
      "            code = generate_response(prompt, max_tokens=150)\n",
```

```
        "            st.code(code, language=\"python\")\n",
        "        else:\n",
        "            st.warning(\"Please input a description.\")\n",
        "\n",
        "elif choice == \"Code Review\":\n",
        "    st.header(\"Automated Code Review\")\n",
        "    code = st.text_area(\"Paste your code here for review:\")\n",
        "    if st.button(\"Review Code\"):\n",
        "        if code.strip():\n",
        "            prompt = f\"Review the following Python code and list any issues or
improvements:\\n\\n{code}\\n\\nReview:\"\n",
        "            review = generate_response(prompt, max_tokens=150)\n",
        "            st.warning(\"Review Comments:\")\n",
        "            st.write(review)\n",
        "        else:\n",
        "            st.warning(\"Please paste code to review.\")\n",
        "\n",
        "elif choice == \"Test Case Generation\":\n",
        "    st.header(\"Generate Test Cases from Requirements\")\n",
        "    req_text = st.text_area(\"Paste the functionality or requirements:\")\n",
        "    if st.button(\"Generate Test Cases\"):\n",
        "        if req_text.strip():\n",
        "            prompt = f\"Based on the following requirements, generate a list of software test
cases:\\n\\n{req_text}\\n\\nTest Cases:\"\n",
        "            cases = generate_response(prompt, max_tokens=150)\n",
        "            st.write(\"Suggested Test Cases:\")\n",
        "            st.write(cases)\n",
        "        else:\n",
        "            st.warning(\"Please input requirements.\")\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "dDdV02HVyGaK",
        "outputId": "a3900bfd-cca4-4768-d0e7-091e95ef2628"
      },
      "execution_count": 26,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Writing Smart_app.py\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "from pyngrok import ngrok\n",
        "import os\n",
        "\n",
        "# Run Streamlit app in background\n",
```

```
        "os.system(\"streamlit run Smart_app.py --server.port 8501 &\")\n",
        "\n",
        "# Wait a bit for the app to start\n",
        "import time\n",
        "time.sleep(5)\n",
        "\n",
        "# Open ngrok tunnel to the Streamlit app\n",
        "public_url = ngrok.connect(8501)\n",
        "print(\"   Your SmartSDLC app is live at:\", public_url)\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "IXqQQx4hzIHn",
        "outputId": "795dd6e7-d246-426f-8d0f-6a064aeed1ca"
      },
      "execution_count": 27,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "   Your SmartSDLC app is live at: NgrokTunnel: \"https://dd40-104-196-97-191.ngrok-free.app\" ->
\"http://localhost:8501\"\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [],
      "metadata": {
        "id": "3Kf3rWAa8ir9"
      },
      "execution_count": null,
      "outputs": []
    }
  ]
}
```