{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": [],

"authorship\_tag": "ABX9TyPmzF7rtq1f6RcMQFI9zsNs",

"include\_colab\_link": true

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "markdown",

"metadata": {

"id": "view-in-github",

"colab\_type": "text"

},

"source": [

"<a href=\"https://colab.research.google.com/github/GeethaDundu/SmartSDLC-AI-Enhanced-Software-Development-Life-Cycle/blob/main/SmartSDLC.ipynb\" target=\"\_parent\"><img src=\"https://colab.research.google.com/assets/colab-badge.svg\" alt=\"Open In Colab\"/></a>"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "yssrJKvJzr-w",

"outputId": "3c5ceaa7-bae5-4c21-8c5f-e56ab15b594c"

},

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\u001b[2K \u001b[90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━\u001b[0m \u001b[32m44.3/44.3 kB\u001b[0m \u001b[31m2.8 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",

"\u001b[2K \u001b[90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━\u001b[0m \u001b[32m10.1/10.1 MB\u001b[0m \u001b[31m63.7 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",

"\u001b[2K \u001b[90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━\u001b[0m \u001b[32m6.9/6.9 MB\u001b[0m \u001b[31m117.0 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",

"\u001b[2K \u001b[90m━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━\u001b[0m \u001b[32m79.1/79.1 kB\u001b[0m \u001b[31m7.2 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",

"\u001b[?25h"

]

}

],

"source": [

"\n",

"!pip install streamlit pyngrok --quiet"

]

},

{

"cell\_type": "code",

"source": [

"!pkill -f ngrok"

],

"metadata": {

"id": "0sM6emUx0DKO"

},

"execution\_count": 2,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"!ngrok authtoken 2yD7JZWRLeAgDLmOWlY3Vi9b0zv\_6NFgPaLQxKbNsCcuJTTmV"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "e\_bFsd0O0H53",

"outputId": "f0ff5ff2-d5a3-4530-b97d-35f8deebd8d5"

},

"execution\_count": 3,

"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": "evIIco8u0NQg",

"outputId": "4ce707ab-a1cc-4c33-9f30-cb147518657d"

},

"execution\_count": 4,

"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": "qL9vgtSr0bMo",

"outputId": "65f57684-8ea6-44e4-e5b0-880e5d987104"

},

"execution\_count": 5,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"🚀 Your SmartSDLC app is live at: NgrokTunnel: \"https://350f-34-16-132-230.ngrok-free.app\" -> \"http://localhost:8501\"\n"

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "iCP9oeTR0kVB"

},

"execution\_count": null,

"outputs": []

}

]

}