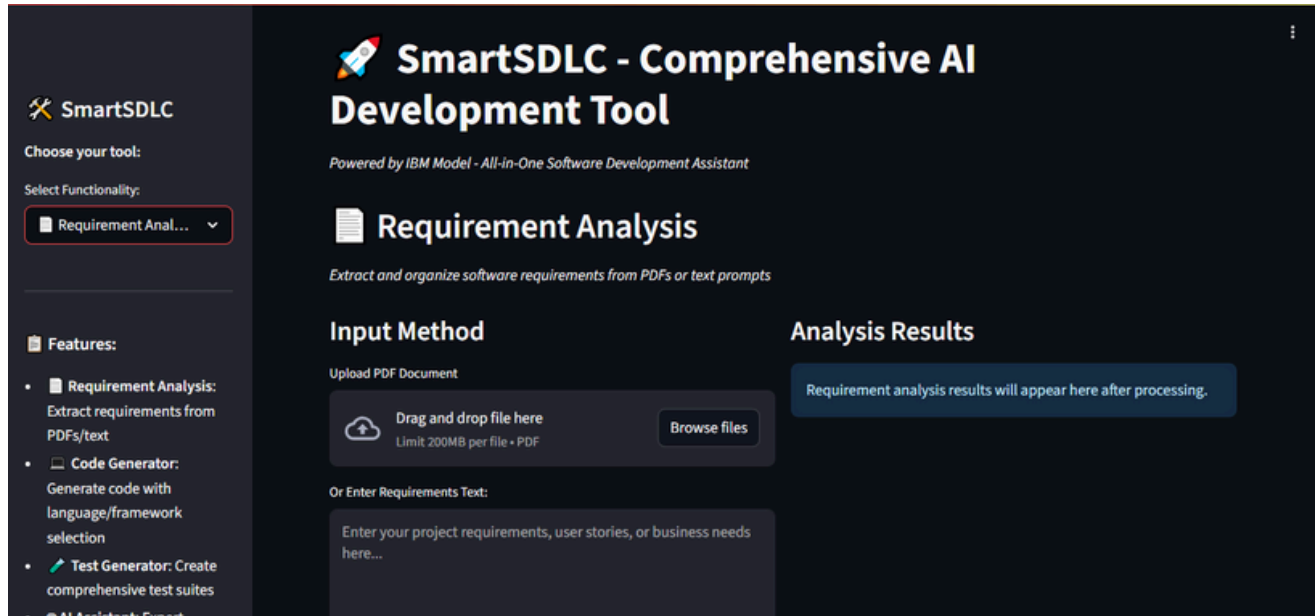


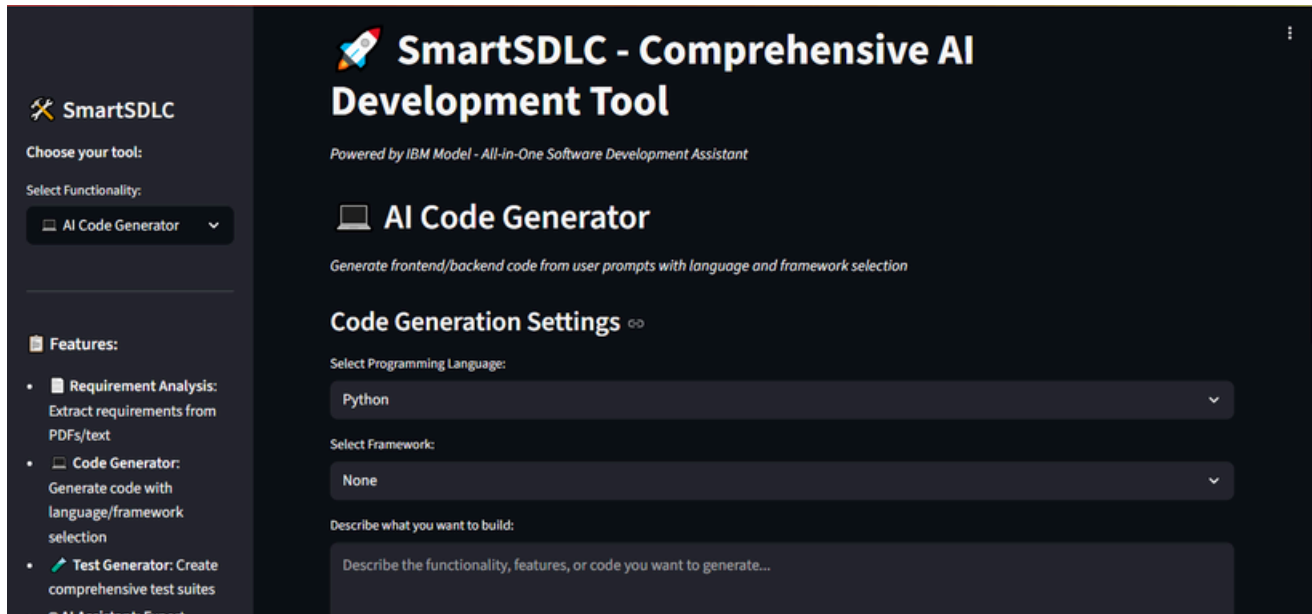
Exploring Application Features: Requirements Analysis



The screenshot shows the 'Requirement Analysis' section of the SmartSDLC tool. The left sidebar has a 'Requirement Anal...' dropdown selected under 'Select Functionality:'. The main area is titled 'Requirement Analysis' and includes a sub-header 'Extract and organize software requirements from PDFs or text prompts'. Under 'Input Method', there is a 'Upload PDF Document' section with a drag-and-drop area (limit 200MB per file - PDF) and a 'Browse files' button. Below this is a text input field for 'Or Enter Requirements Text:'. The 'Analysis Results' section on the right contains a message: 'Requirement analysis results will appear here after processing.'

Description: The user uploads a business requirements document in PDF format or types a detailed text description of the software idea. This input typically contains objectives, features, and system expectations. It may be technical or non-technical in tone. The system accepts both structured and unstructured text.

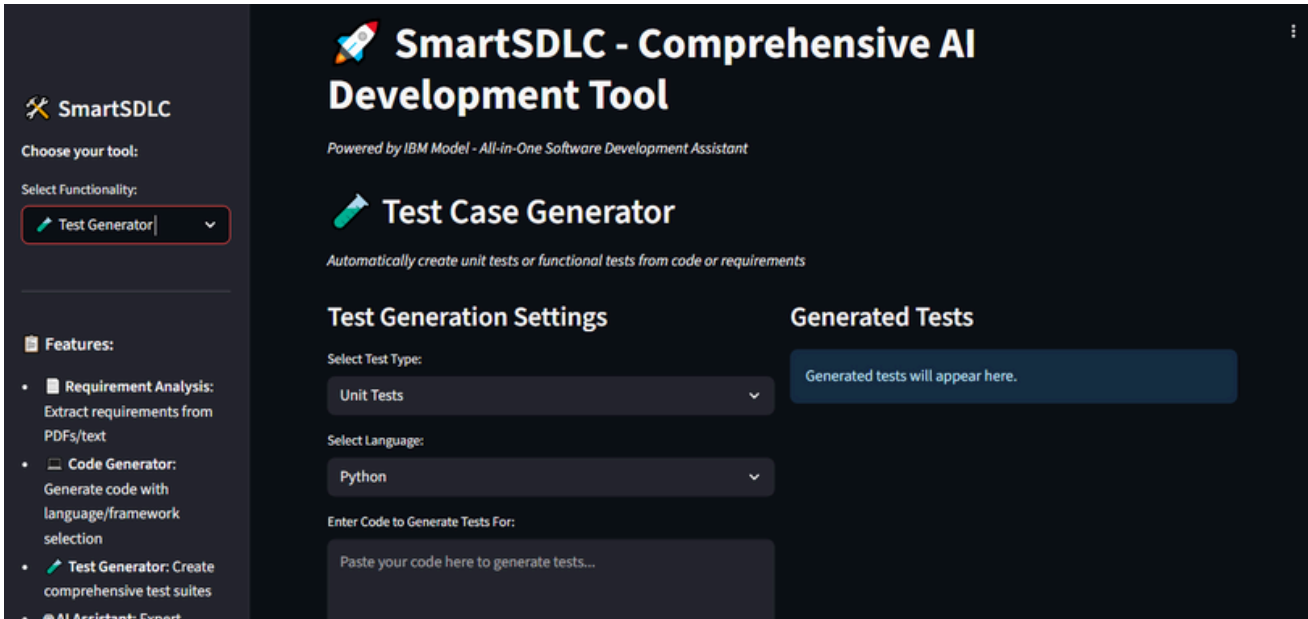
AI Code Generator



The screenshot shows the 'AI Code Generator' section of the SmartSDLC tool. The left sidebar has an 'AI Code Generator' dropdown selected under 'Select Functionality:'. The main area is titled 'AI Code Generator' and includes a sub-header 'Generate frontend/backend code from user prompts with language and framework selection'. Under 'Code Generation Settings', there are two dropdown menus: 'Select Programming Language:' (set to 'Python') and 'Select Framework:' (set to 'None'). Below these is a text input field for 'Describe what you want to build:'. The input field contains the placeholder text: 'Describe the functionality, features, or code you want to generate...'

Description: The user provides a high-level prompt describing the desired functionality of an application or module. They also select a programming language (e.g., Python, JavaScript) and a framework (e.g., FastAPI, React). The input should clearly mention core features such as login, CRUD operations, or dashboards.

Test Case Generator



SmartSDLC - Comprehensive AI Development Tool
Powered by IBM Model - All-in-One Software Development Assistant

Test Case Generator
Automatically create unit tests or functional tests from code or requirements

Choose your tool:
Select Functionality:
Test Generator

Features:

- Requirement Analysis:** Extract requirements from PDFs/text
- Code Generator:** Generate code with language/framework selection
- Test Generator:** Create comprehensive test suites
- AI Assistant:** Expert

Test Generation Settings

Select Test Type:
Unit Tests

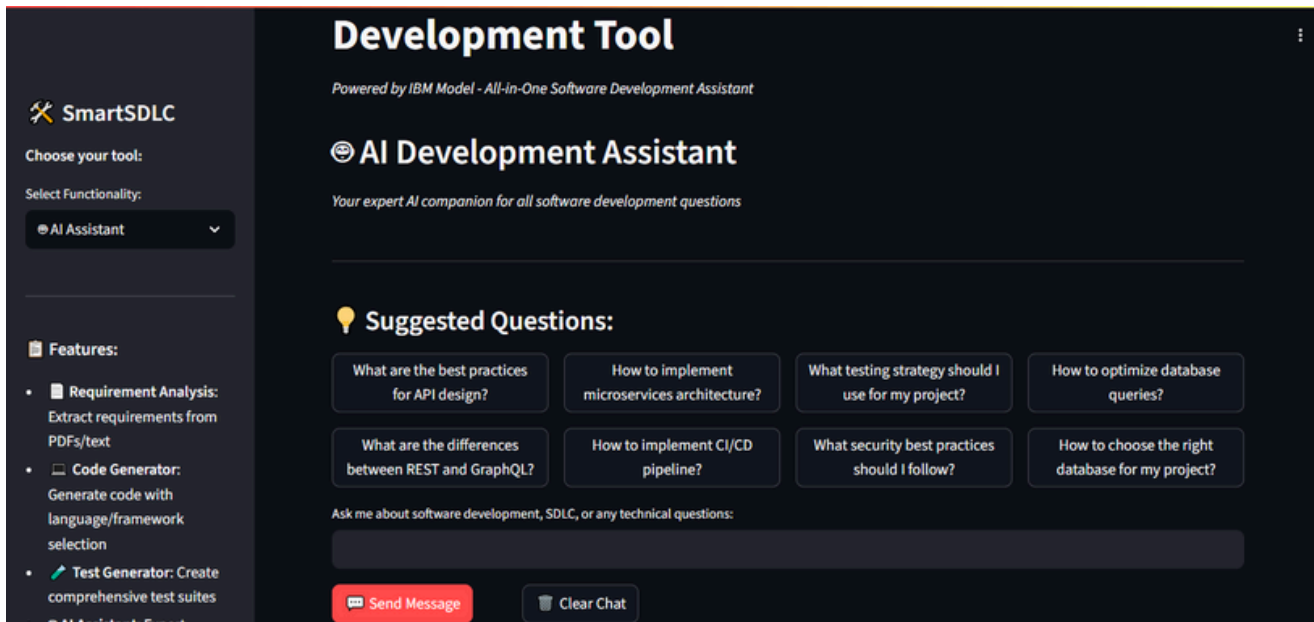
Select Language:
Python

Enter Code to Generate Tests For:
Paste your code here to generate tests...

Generated Tests
Generated tests will appear here.

Description: The input can either be raw source code or a prompt describing what needs to be tested. The user then selects the type of test: Unit, Integration, or API. The input should specify modules, endpoints, or behavior that requires validation. Optionally, users can mention edge cases or libraries.

AI Chat Assistant



SmartSDLC - Comprehensive AI Development Tool
Powered by IBM Model - All-in-One Software Development Assistant

AI Development Assistant
Your expert AI companion for all software development questions

Choose your tool:
Select Functionality:
AI Assistant

Features:

- Requirement Analysis:** Extract requirements from PDFs/text
- Code Generator:** Generate code with language/framework selection
- Test Generator:** Create comprehensive test suites
- AI Assistant:** Expert

Suggested Questions:

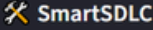
- What are the best practices for API design?
- How to implement microservices architecture?
- What testing strategy should I use for my project?
- How to optimize database queries?
- What are the differences between REST and GraphQL?
- How to implement CI/CD pipeline?
- What security best practices should I follow?
- How to choose the right database for my project?

Ask me about software development, SDLC, or any technical questions:

Send Message Clear Chat

Description: The user enters a free-form query related to software development, system architecture, testing practices, or SDLC methodologies. Questions can be practical (“How do I deploy a FastAPI app?”) or conceptual (“What is the difference between unit and integration testing?”). Natural language input is supported.

Requirement Analysis Output



Choose your tool:

Select Functionality:

Requirement Anal...

Features:

- Requirement Analysis: Extract requirements from PDFs/text
- Code Generator: Generate code with language/framework selection
- Test Generator: Create comprehensive test suites
- AI Assistant: Expert

MODEL Info:

Model: ibm-granite/granite-3.3-2b-instruct

Status: Connected

Requirement Analysis

Extract and organize software requirements from PDFs or text prompts

Input Method

Upload PDF Document

Drag and drop file here
Limit 200MB per file • PDF

Browse files

Or Enter Requirements Text:

Create a mobile application for a food delivery service that supports order placement, real-time tracking, and customer reviews. The app should have an admin dashboard to manage restaurants, menus, and orders. Ensure user authentication and payment integration.

Analyze Requirements

Analysis Results

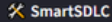
Functional Requirements:

- Order Placement:** Enable users to browse menus, select items, and place orders with ease.
- Real-Time Tracking:** Provide GPS-based tracking for customers to monitor order status and delivery progress.
- Customer Reviews:** Allow users to rate and review restaurants and delivery experiences.
- Admin Dashboard:** Enable admins to manage restaurant profiles, update menus, and monitor orders.
- User Authentication:** Implement secure login/signup with multi-factor authentication (MFA).
- Payment Integration:** Support seamless payment processing via credit/debit cards, digital wallets, and other methods.

Non-Functional Requirements:

Description: The system extracted clear Functional Requirements such as order placement, tracking, and user authentication, along with Non-Functional Requirements like security and performance. It also generated Technical Requirements specifying an admin dashboard and payment integration. Additionally, Agile-style User Stories were auto-generated for product backlog use.

AI Code Generator Output



Choose your tool:

Select Functionality:

AI Code Generator

Features:

- Requirement Analysis: Extract requirements from PDFs/text
- Code Generator: Generate code with language/framework selection
- Test Generator: Create comprehensive test suites
- AI Assistant: Expert development guidance

MODEL Info:

Model: ibm-granite/granite-3.3-2b-instruct

Status: Connected

AI Code Generator

Generate frontend/backend code from user prompts with language and framework selection

Code Generation Settings

Select Programming Language:

Python

Select Framework:

FastAPI

Describe what you want to build:

Build a RESTful API for a task management system using FastAPI. It should allow users to register, login, create tasks, update status, and delete tasks. Include token-based authentication and error handling.

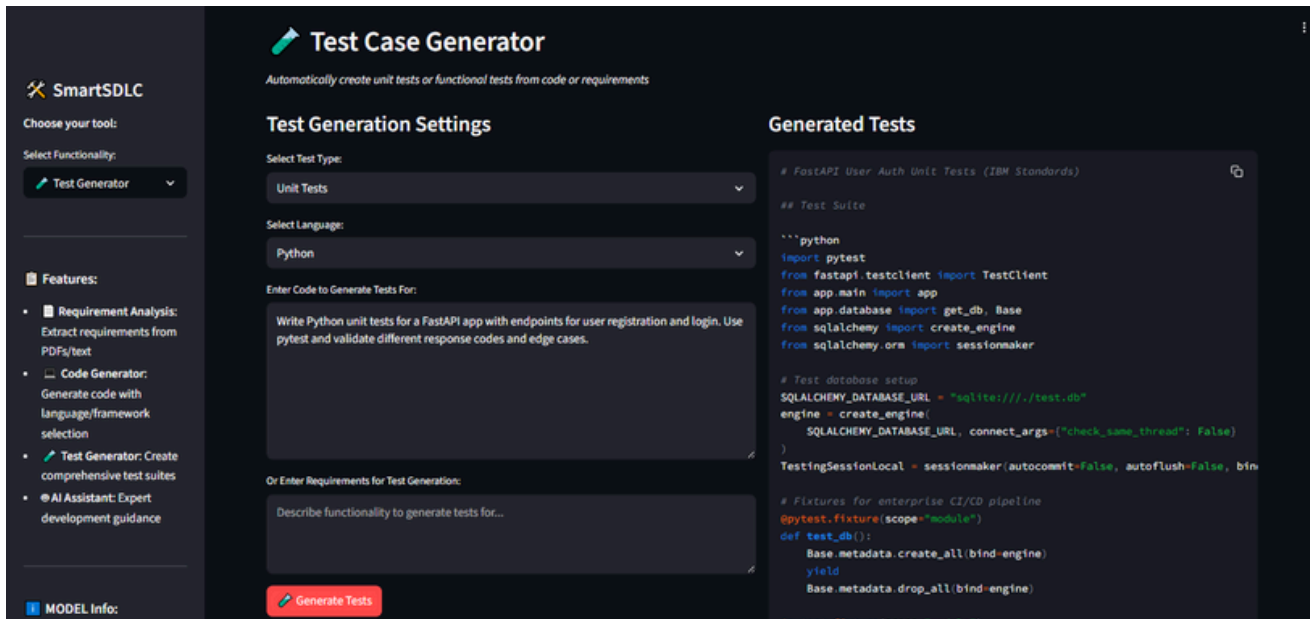
Generate Code

Generated Code

```
'''python
from fastapi import FastAPI, HTTPException, Depends, status
from pydantic import BaseModel
from typing import List, Optional
from datetime import datetime
from fastapi.security import OAuth2PasswordBearer, OAuth2PasswordRequestForm
from jose import JWTError, jwt
from passlib.context import CryptContext
import uuid
```

Description: The Granite model returned clean, well-structured FastAPI backend code for a task management system. It included endpoints for user registration, login, and task CRUD operations. Token-based authentication was implemented using JWT. Each function was clearly commented, and the entire codebase was formatted for direct use.

Test Case Generator Output



Test Case Generator
 Automatically create unit tests or functional tests from code or requirements

Test Generation Settings

Select Test Type: Unit Tests

Select Language: Python

Enter Code to Generate Tests For:

Write Python unit tests for a FastAPI app with endpoints for user registration and login. Use pytest and validate different response codes and edge cases.

Or Enter Requirements for Test Generation:

Describe functionality to generate tests for...

Generated Tests

```
# FastAPI User Auth Unit Tests (IBM Standards)
## Test Suite

'''python
import pytest
from fastapi.testclient import TestClient
from app.main import app
from app.database import get_db, Base
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker

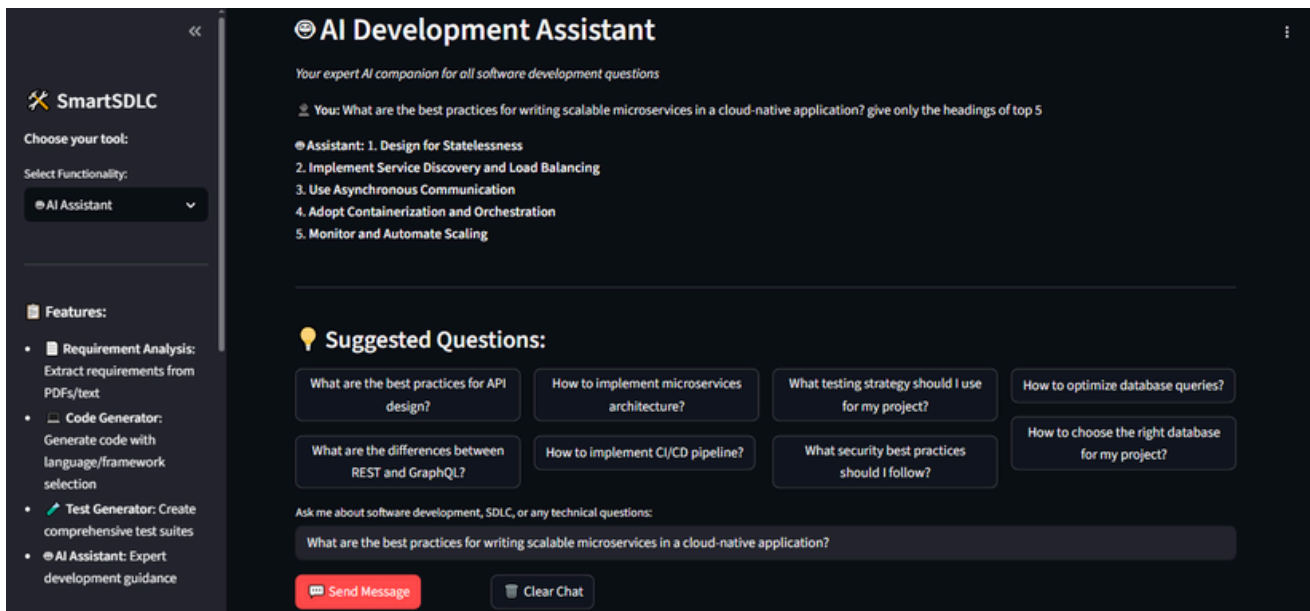
# Test database setup
SQLALCHEMY_DATABASE_URL = "sqlite:///./test.db"
engine = create_engine(
    SQLALCHEMY_DATABASE_URL, connect_args={"check_same_thread": False})
TestingSessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)

# Fixtures for enterprise CI/CD pipeline
@pytest.fixture(scope="module")
def test_db():
    Base.metadata.create_all(bind=engine)
    yield
    Base.metadata.drop_all(bind=engine)

@pytest.fixture(scope="function")
def client():
    client = TestClient(app)
    return client'''
```

Description: The system produced a set of pytest-based unit tests targeting user registration and login endpoints. Each test function included edge cases (e.g., invalid credentials) and status code assertions. The output followed best practices for API testing and ensured proper coverage of core authentication logic.

AI Chat Assistant Output



AI Development Assistant
 Your expert AI companion for all software development questions

You: What are the best practices for writing scalable microservices in a cloud-native application? give only the headings of top 5

Assistant: 1. Design for Statelessness
 2. Implement Service Discovery and Load Balancing
 3. Use Asynchronous Communication
 4. Adopt Containerization and Orchestration
 5. Monitor and Automate Scaling

Suggested Questions:

- What are the best practices for API design?
- How to implement microservices architecture?
- What testing strategy should I use for my project?
- How to optimize database queries?
- What are the differences between REST and GraphQL?
- How to implement CI/CD pipeline?
- What security best practices should I follow?
- How to choose the right database for my project?

Ask me about software development, SDLC, or any technical questions:

What are the best practices for writing scalable microservices in a cloud-native application?

Send Message **Clear Chat**

Description: The assistant responded with enterprise-grade recommendations for designing scalable microservices. This included API gateway usage, containerization with Docker, service discovery, asynchronous messaging patterns, and observability tools. The answer was broken down into best practices and included architecture diagrams and code snippets where applicable.

Project Implementation

```

# --- 1. Create SmartSDLC app with IBM Granite ---
print("Creating SmartSDLC app with IBM Granite model...")

Uninstalling nvidia-nvtx-cu12-12.4.127:
Successfully uninstalled nvidia-nvtx-cu12-12.4.127
Attempting uninstall: nvidia-nvjitlink-cu12
Found existing installation: nvidia-nvjitlink-cu12 12.5.82
Uninstalling nvidia-nvjitlink-cu12-12.5.82:
Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
Attempting uninstall: nvidia-nccl-cu12
Found existing installation: nvidia-nccl-cu12 2.21.5
Uninstalling nvidia-nccl-cu12-2.21.5:
Successfully uninstalled nvidia-nccl-cu12-2.21.5
Attempting uninstall: nvidia-curand-cu12
Found existing installation: nvidia-curand-cu12 10.3.6.82
Uninstalling nvidia-curand-cu12-10.3.6.82:
Successfully uninstalled nvidia-curand-cu12-10.3.6.82
Attempting uninstall: nvidia-cuda-runtime-cu12
Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
Attempting uninstall: nvidia-cuda-nvrtc-cu12
Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
Attempting uninstall: nvidia-cuda-cupti-cu12
Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
Attempting uninstall: nvidia-cublas-cu12
Found existing installation: nvidia-cublas-cu12 12.5.3.2
Uninstalling nvidia-cublas-cu12-12.5.3.2:
Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
Attempting uninstall: nvidia-cusparse-cu12
Found existing installation: nvidia-cusparse-cu12 12.5.1.3
Uninstalling nvidia-cusparse-cu12-12.5.1.3:
Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
Attempting uninstall: nvidia-cufft-cu12
Found existing installation: nvidia-cufft-cu12 11.2.3.61
Uninstalling nvidia-cufft-cu12-11.2.3.61:
Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
Attempting uninstall: nvidia-cudnn-cu12
Found existing installation: nvidia-cudnn-cu12 9.3.0.75
Uninstalling nvidia-cudnn-cu12-9.3.0.75:
Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
Attempting uninstall: nvidia-cusolver-cu12
Found existing installation: nvidia-cusolver-cu12 11.6.3.83
Uninstalling nvidia-cusolver-cu12-11.6.3.83:
Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Attempting uninstall: torch
Found existing installation: torch 2.6.0+cu124
Uninstalling torch-2.6.0+cu124:
Successfully uninstalled torch-2.6.0+cu124

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source o
torchvision 0.21.0+cu124 requires torch==2.6.0, but you have torch 2.7.1 which is incompatible.
torchaudio 2.6.0+cu124 requires torch==2.6.0, but you have torch 2.7.1 which is incompatible.
fastai 2.7.19 requires torch<2.7,>=1.10, but you have torch 2.7.1 which is incompatible.
Successfully installed nvidia-cublas-cu12-12.5.4.1 nvidia-cuda-cupti-cu12-12.6.80 nvidia-cuda-nvrtc-cu12-12.6.77 nvidia-cuda-runtime-cu12
Creating SmartSDLC app with IBM Granite model...

st.session_state.chat_history.append({"role": "assistant", "content": response})

# Clear temp input
if 'temp_input' in st.session_state:
    del st.session_state.temp_input

st.rerun()

# Sidebar information
st.sidebar.markdown("----")
st.sidebar.markdown("### 📄 Features:")
st.sidebar.markdown("
- **📄 Requirement Analysis**: Extract requirements from PDFs/text
- **🔧 Code Generator**: Generate code with language/framework selection
- **🧪 Test Generator**: Create comprehensive test suites
- **🤖 AI Assistant**: Expert development guidance
")

st.sidebar.markdown("----")
st.sidebar.markdown("### 📘 MODEL Info:")
st.sidebar.markdown("**Model:** ibm-granite/granite-3.3-2b-instruct")
st.sidebar.markdown("**Status:** 🟢 Connected")

st.sidebar.markdown("----")
st.sidebar.markdown("### 🔒 Privacy:")
st.sidebar.markdown("
- IBM Model Successfully Integrated
- No data is stored permanently
- Sessions are isolated
")

# Footer
st.markdown("----")
st.markdown("**Built with Streamlit and IBM Granite Model - Comprehensive Software Development Tool**")

print("✅ SmartSDLC app with IBM Granite model created successfully!")

# --- 2. Start Streamlit with ngrok ---

```

```

print("
• 📄 PDF/Prompt-Based Requirement Analysis
• 🧪 AI Code Generator (Multiple Languages & Frameworks)
• 🧪 Test Case Generator (Unit, Integration, Functional)
• 🤖 AI Development Assistant
")
print("----")
print("🔍 Using IBM Granite Model (Local - No API needed)")
print("⚡ Fast responses with local model inference")
print("🛑 Press Ctrl+C to stop the application")

# Keep the app running
try:
    while True:
        time.sleep(1)
except KeyboardInterrupt:
    print("\n🛑 Stopping the app...")
    ngrok.disconnect(public_url)
    ngrok.kill()
    print("✅ App stopped successfully")

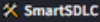
except Exception as e:
    print(f"❌ Error starting ngrok tunnel: {e}")
    print("Possible solutions:")
    print("1. Check your internet connection")
    print("2. Verify your ngrok auth token")
    print("3. Make sure port 8501 is not already in use")
    print("4. Try restarting the kernel and running again")

***
✅ Ngrok token set successfully
Starting SmartSDLC app with IBM Granite model...
Waiting for Streamlit to initialize...
🟢 SmartSDLC with IBM Granite is now running!
🌐 Public URL: NgrokTunnel: "https://da36-34-98-89-210.ngrok-free.app" -> "http://localhost:8501"
👉 Click the link above to access your SmartSDLC app!
📱 The app is mobile-friendly and works on all devices

***
All Features Available:
• 📄 PDF/Prompt-Based Requirement Analysis
• 🧪 AI Code Generator (Multiple Languages & Frameworks)
• 🧪 Test Case Generator (Unit, Integration, Functional)
• 🤖 AI Development Assistant

***
🔍 Using IBM Granite Model (Local - No API needed)
⚡ Fast responses with local model inference
🛑 Press Ctrl+C to stop the application

```


 SmartSDLC

Choose your tool:

Select Functionality:

Requirement A...

Features:

- Requirement Analysis: Extract requirements from PDFs/text
- Code Generator: Generate code with language/framework selection
- Test Generator: Create comprehensive test suites
- AI Assistant: Expert development guidance


MODEL Info:

Model: ibm-granite/granite-3.3-2b-instruct

Status: Connected

Privacy:

- IBM Model Successfully Integrated
- No data is stored permanently
- Sessions are isolated

 SmartSDLC - Comprehensive AI Development Tool

Powered by IBM Model - All-in-One Software Development Assistant

Requirement Analysis

Extract and organize software requirements from PDFs or text prompts

Input Method

Upload PDF Document

Drag and drop file here
Limit: 200MB per file - PDF

Browse files

Or Enter Requirements Text:

Enter your project requirements, user stories, or business needs here...

Analyze Requirements

Analysis Results

Requirement analysis results will appear here after processing.

Built with Streamlit and IBM Granite Model - Comprehensive Software Development Tool