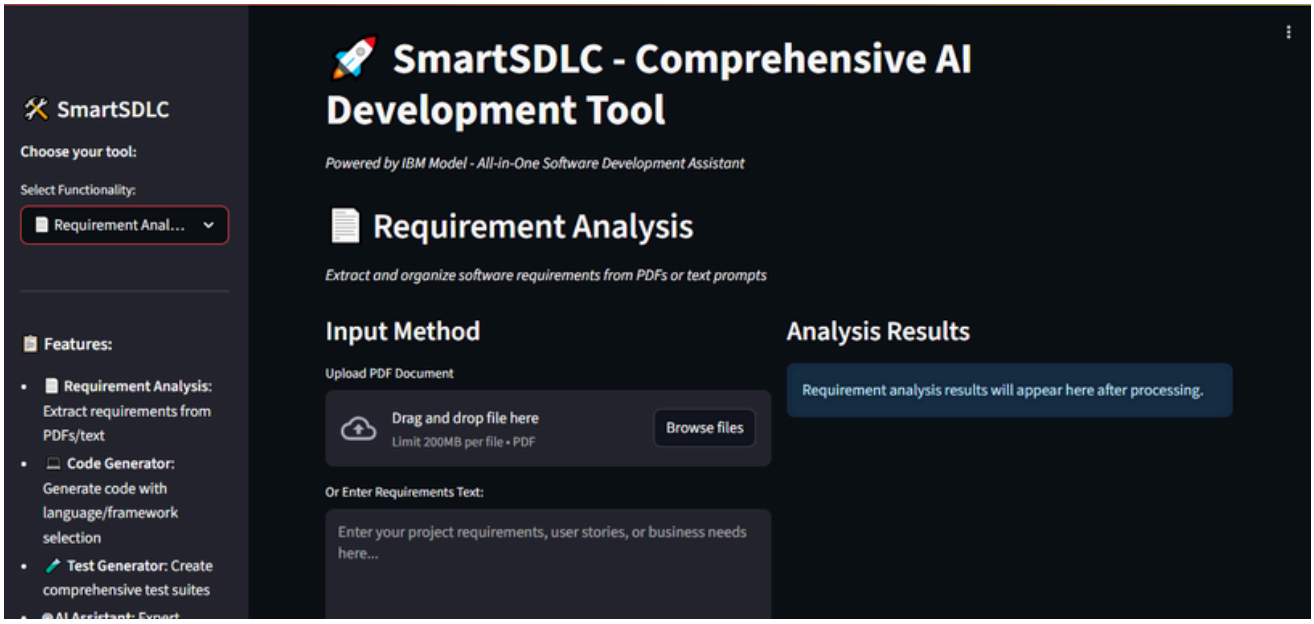
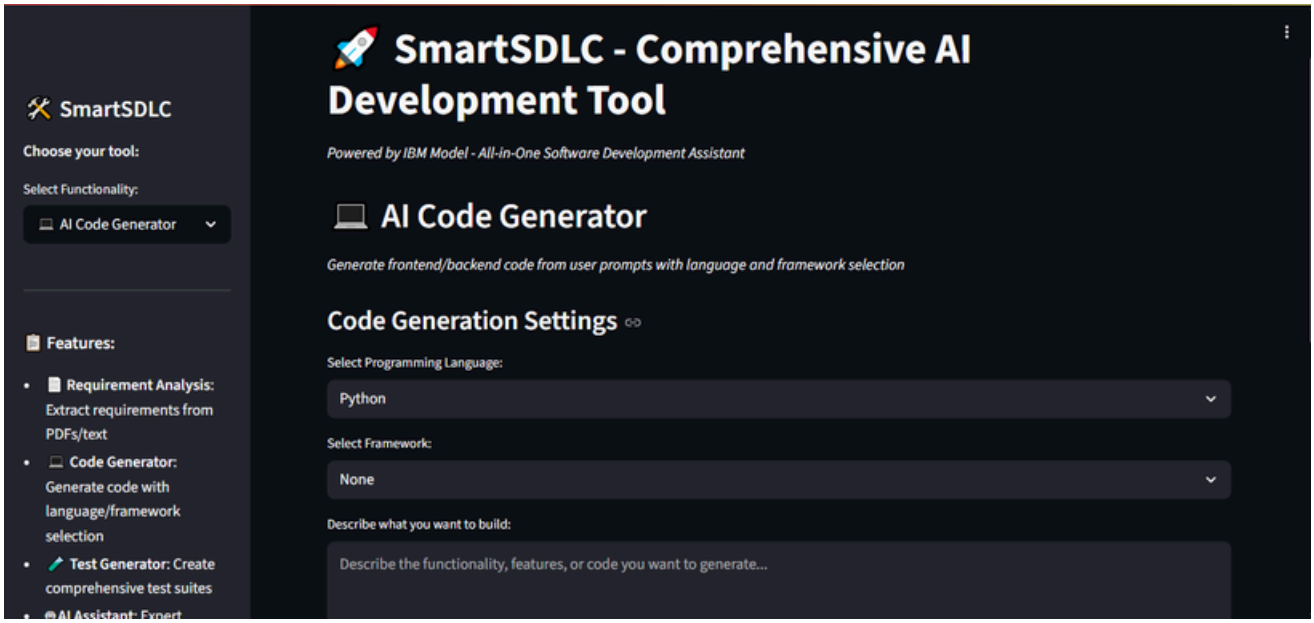
Exploring Application Features:

Requirements Analysis



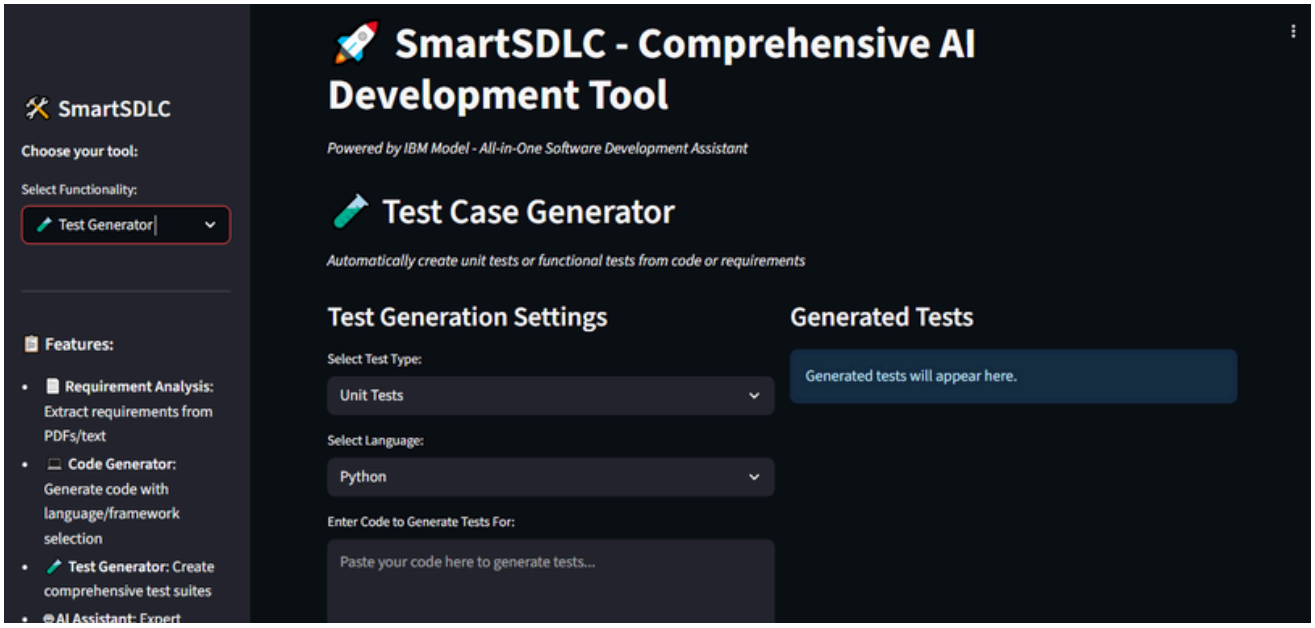
Description: The user uploads a business requirements document in PDF format or types a detailed text description of the software idea. This input typically contains objectives, features, and system expectations. It may be technical or non-technical in tone. The system accepts both structured and unstructured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

AI Code Generator



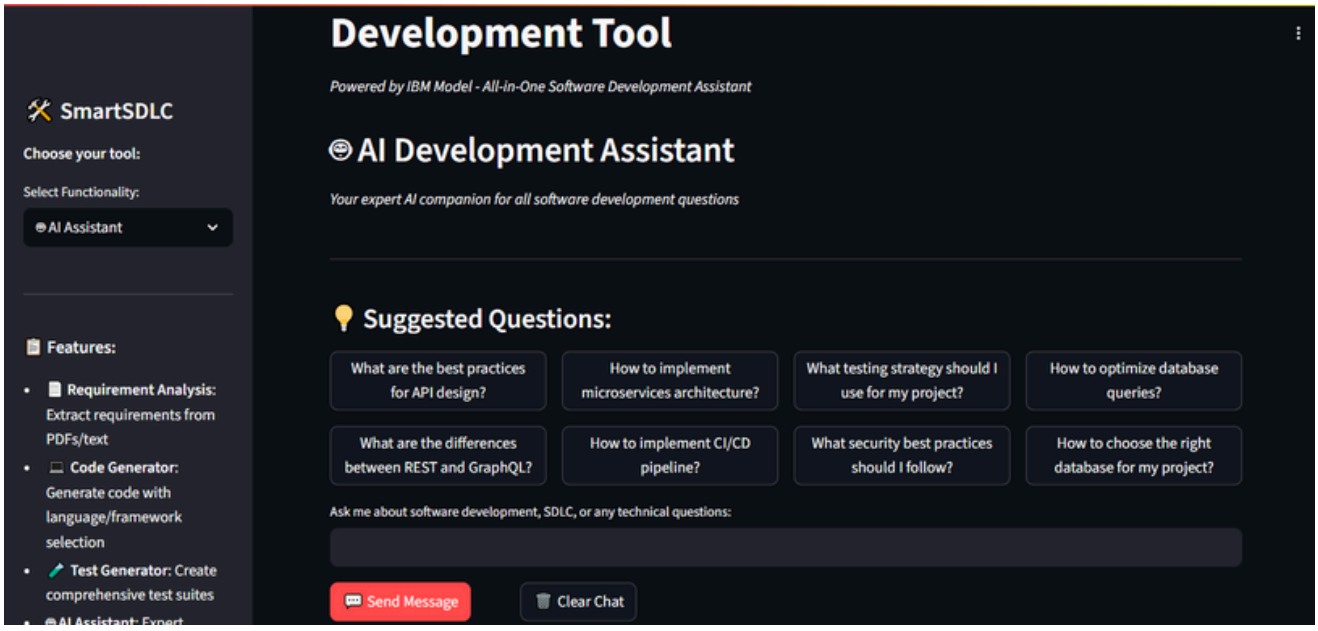
Description: The user provides a high-level prompt describing the desired functionality of an application or module. They also select a programming language (e.g., Python, JavaScript) and a framework (e.g., FastAPI, React). The input should clearly mention core features such as login, CRUD operations,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

Test Case Generator



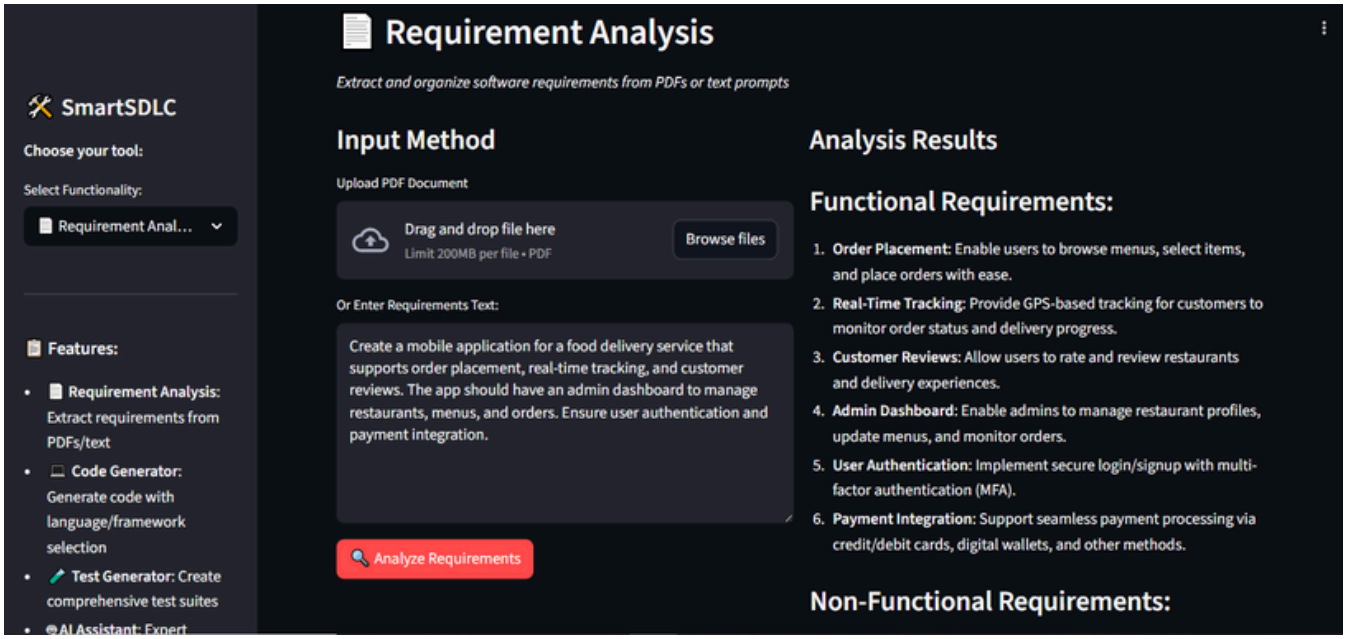
Description: The input can either be raw source code or a prompt describing what needs to be tested. The user then selects the type of test: Unit, Integration, or API. The input should specify modules, endpoints, or behavior that requires validation. Optionally, users can mention edge cases or libraries.

AI Chat Assistant



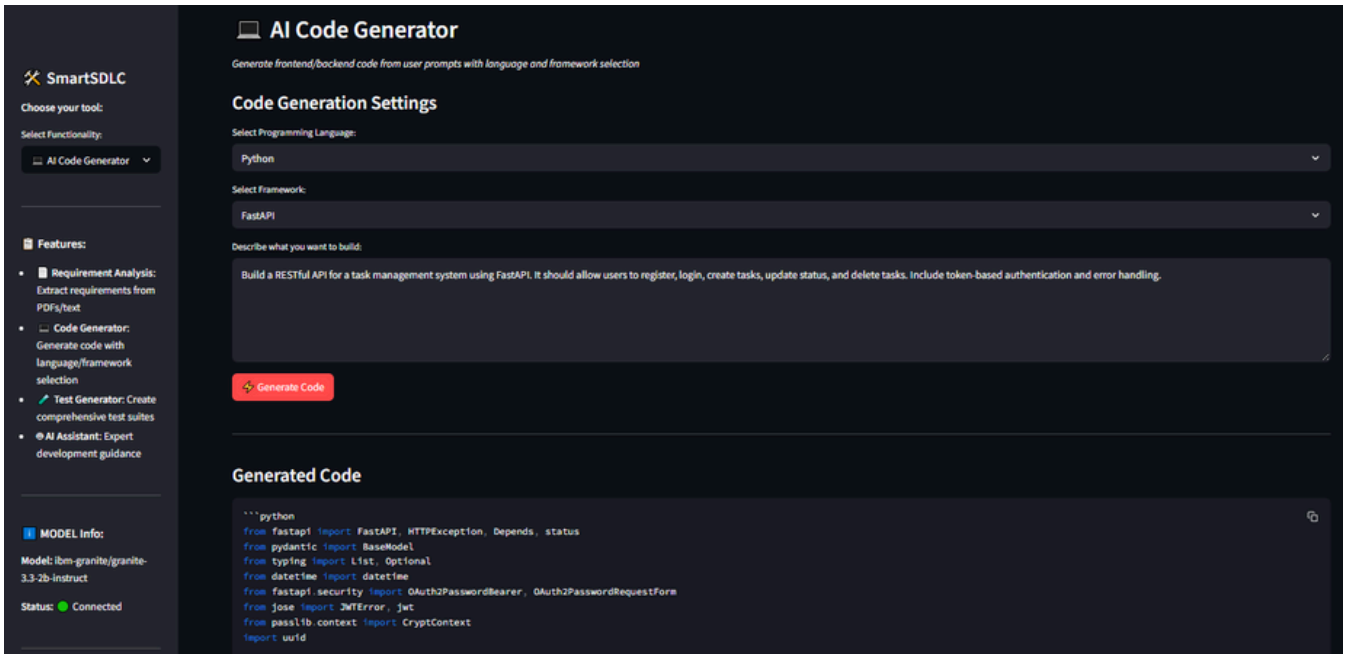
Description: The user enters a free-form query related to software development, system architecture, testing practices, or SDLC methodologies. Questions can be practical (“How do I deploy a FastAPI app?”) or conceptual (“What is the difference between unit and integration testing?”). Natural language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

Requirement Analysis Output



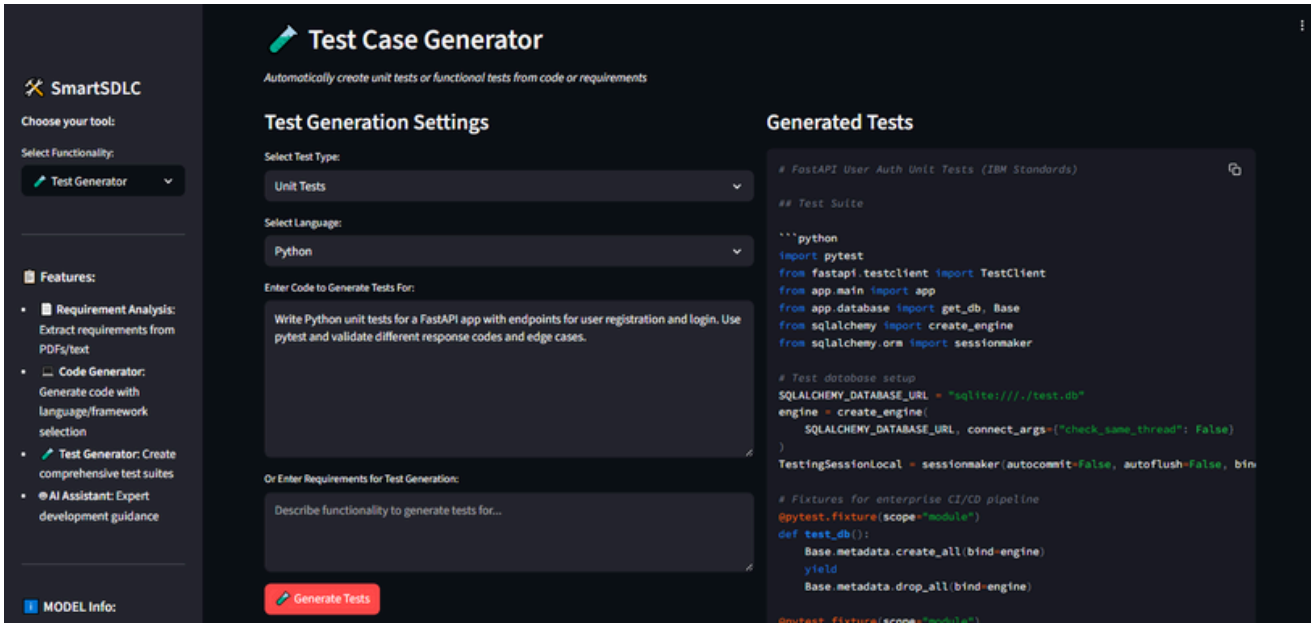
Description: The system extracted clear Functional Requirements such as order placement, tracking, and user authentication, along with Non-Functional Requirements like security and performance. It also generated Technical Requirements specifying an admin dashboard and payment integration. Additionally,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

AI Code Generator Output



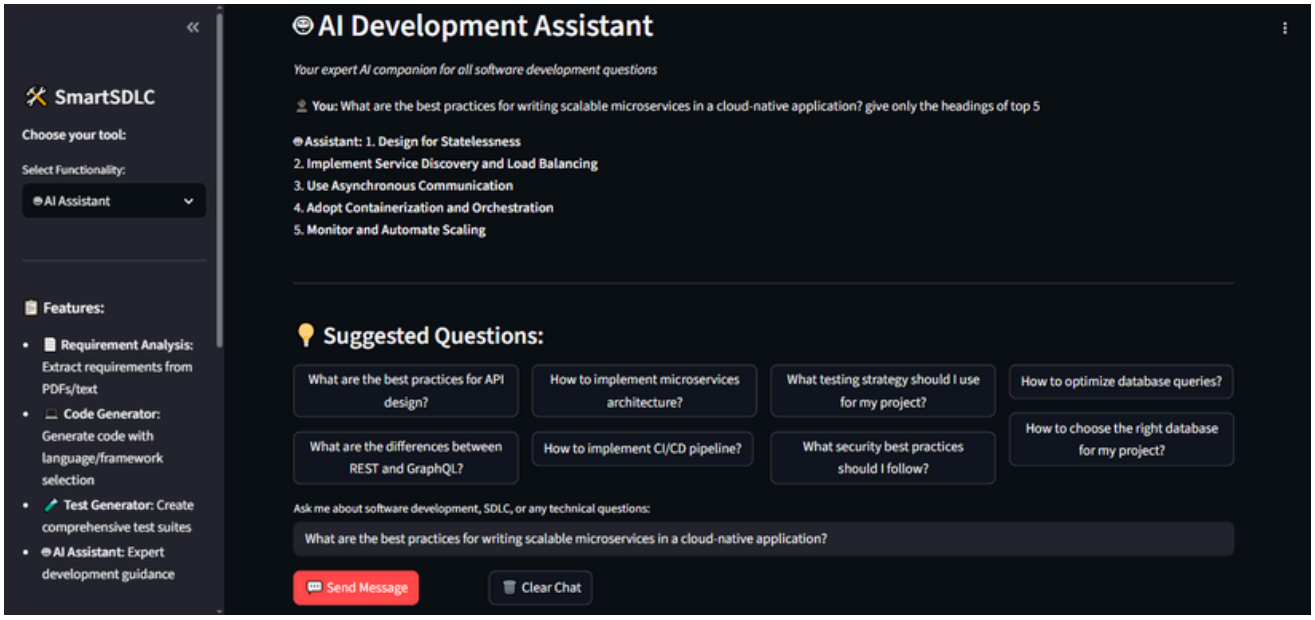
Description: The Granite model returned clean, well-structured FastAPI backend code for a task management system. It included endpoints for user registration, login, and task CRUD operations. Token-based authentication was implemented using JWT. Each function was clearly commented, and the entire codebase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

Test Case Generator Output



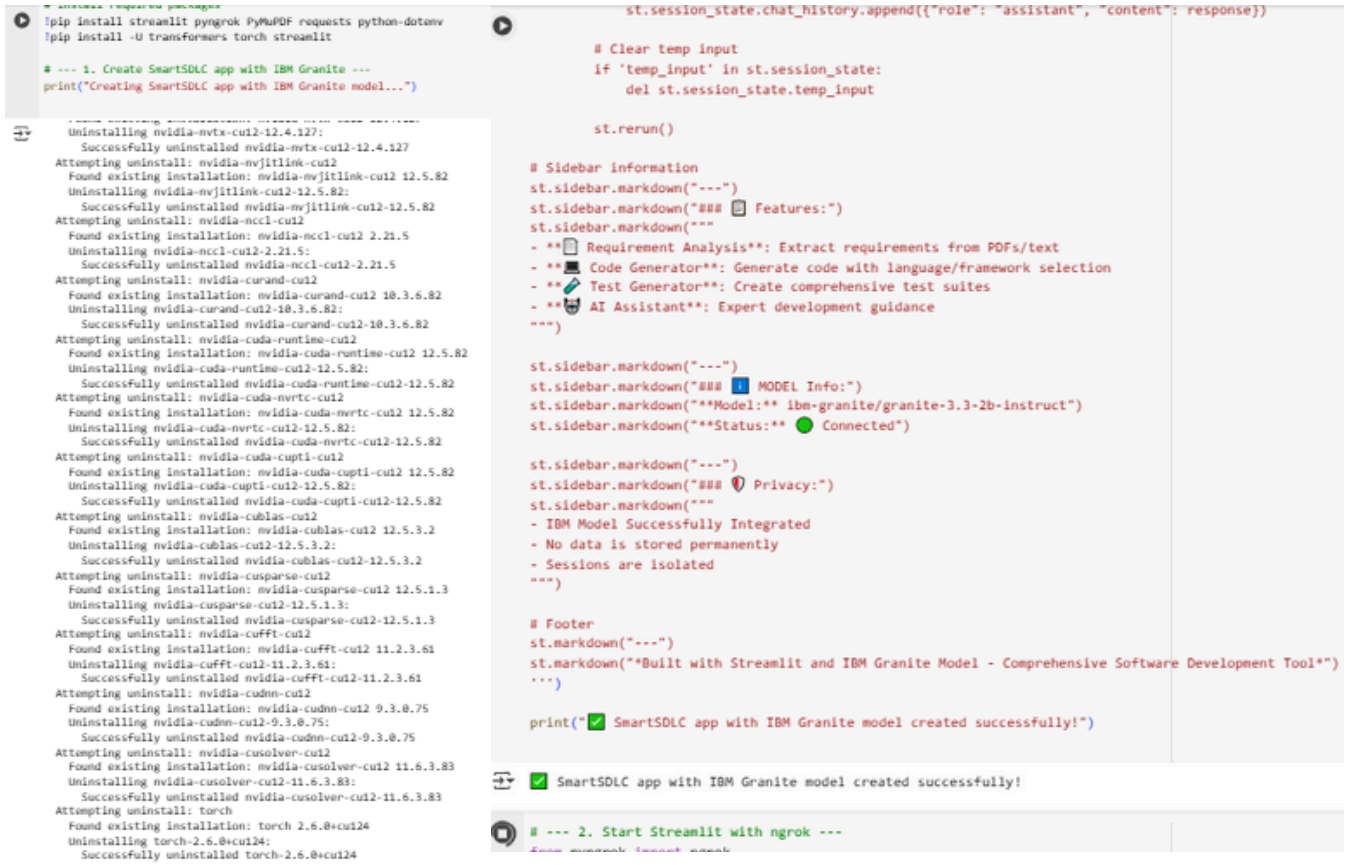
Description: The system produced a set of pytest-based unit tests targeting user registration and login endpoints. Each test function included edge cases (e.g., invalid credentials) and status code assertions. The output followed best practices for API testing and ensured proper coverage of core authentication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

AI Chat Assistant Output



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



```
!pip install streamlit pyngrok PyMuPDF requests python-dotenv
!pip install -U transformers torch streamlit

# --- 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-cu12-12.6.4.1 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("### ℹ️ 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!")
```

```
✅ SmartSDLC app with IBM Granite model created successfully!
```

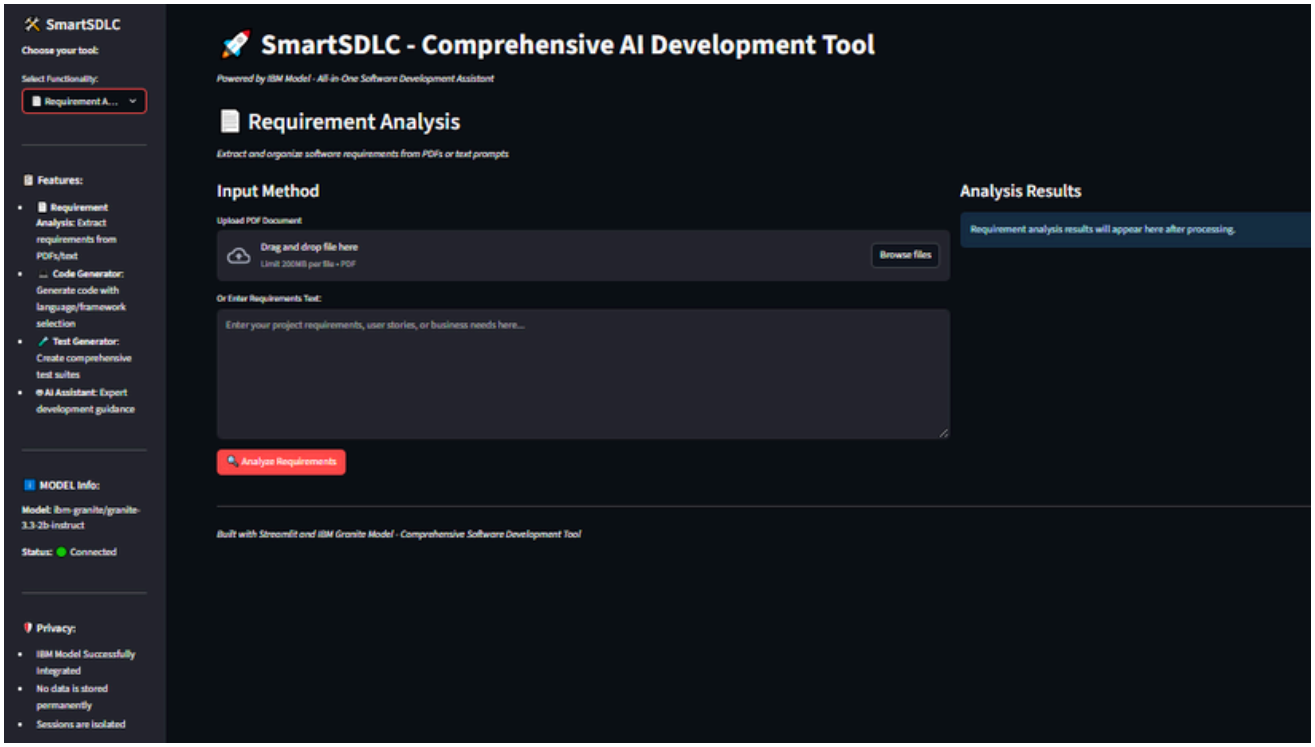
```
# --- 2. Start Streamlit with ngrok ---
```

```
    print("   • 📄 PDF/Prompt-Based Requirement Analysis")
    print("   • 💻 AI Code Generator (Multiple Languages & Frameworks)")
    print("   • 🧪 Test Case Generator (Unit, Integration, Functional)")
    print("   • 🤖 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...
\n✅ SmartSDLC with IBM Granite is now running!
🌐 Public URL: NgrokTunnel: "https://da36-34-90-89-210.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