

# SMARTSDLC: AI-Enhanced Software Development Lifecycle

## Project Documentation

Team id: LTVIP2025TMID22707

Team Size:4

Team Leader:Katta Karthik

Team Member:K Sai Gopala Krishna

Team Member:K Sri Charan

Team Member:Jetipiti Venu

### **1. Introduction**

#### **Project Overview:**

An AI-driven platform for real-time language correction and learning, leveraging IBM's Granite model to:

- ✓ Fix grammar/spelling errors
- ✓ Detect language automatically
- ✓ Explain corrections pedagogically

#### **Target Users:**

- Language learners
- Educators
- Content creators

## 2. Features

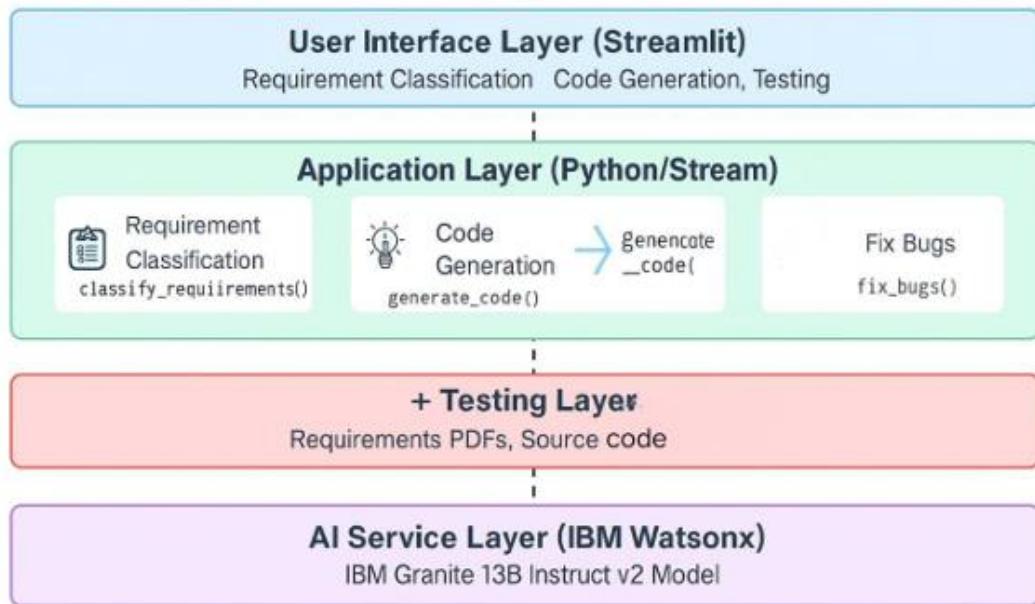
Feature	Description
<b>Grammar Correction</b>	Fixes errors with explanations (e.g., verb tense)
<b>Language Detection</b>	Identifies 50+ languages
<b>Learning Dashboard</b>	Tracks progress over time

## 3. Architecture:

SmartSDLC follows a clean, modular architecture:

1. **\*\*User Interface (Streamlit)\*\*:** Collects input, displays results, and manages interactions.
2. **\*\*Application Logic (Python)\*\*:** Processes user commands, forms AI prompts, and handles session state.
3. **\*\*AI Service Layer (IBM Watsonx)\*\*:** IBM Granite 3.3 Instruct model generates context-aware outputs.
4. **\*\*Temporary State Memory\*\*:** User session and intermediate data stored in-memory using Streamlit session state.

## SmartSDLC - Architecture Diagram



## 4. Setup Instructions

**\*\*Prerequisites\*\*:** - Python 3.8 or above - IBM Cloud account with Watsonx access - Streamlit, pandas, python-dotenv, ibm-watsonx-ai

### **\*\*Installation Steps\*\*:**

1. Clone the project and navigate to the folder
2. Create a virtual environment: `python -m venv venv`
3. Activate the virtual environment: - Windows:  
`.\venv\Scripts\activate`
4. Install dependencies: `pip install -r requirements.txt`

5. Create ` `.env` file with the following:

IBM\_API\_KEY="your\_key" PROJECT\_ID="your\_project\_id"  
BASE\_URL=<https://eu-de.ml.cloud.ibm.com>

6. Run the app: `streamlit run SMART\_SDLC.py`

## 5. API Documentation

SmartSDLC does not expose traditional REST APIs but uses IBM Watsonx's `generate\_text()` method via the Python SDK.

**\*\*Example Prompt Usage\*\*:** - Code Generation: "Generate Python code that implements a login system."

- Bug Fixing: "Fix the following code: [BUGGY CODE]"
  - Testing: "Write pytest unit tests for the given function."
- \*\*AI Parameters\*\*:** - max\_new\_tokens = 500
- temperature = 0.7
  - top\_p = 1.0
  - decoding\_method = sample

## 6. Authentication

IBM Watsonx is accessed using secure API key authentication. Credentials are stored in a ` `.env` file and loaded using `python-dotenv`.

**\*\*Security Practice\*\*:**

- Do not hardcode API keys.
- Use ` `.env` and ` `.gitignore` to prevent accidental exposure.

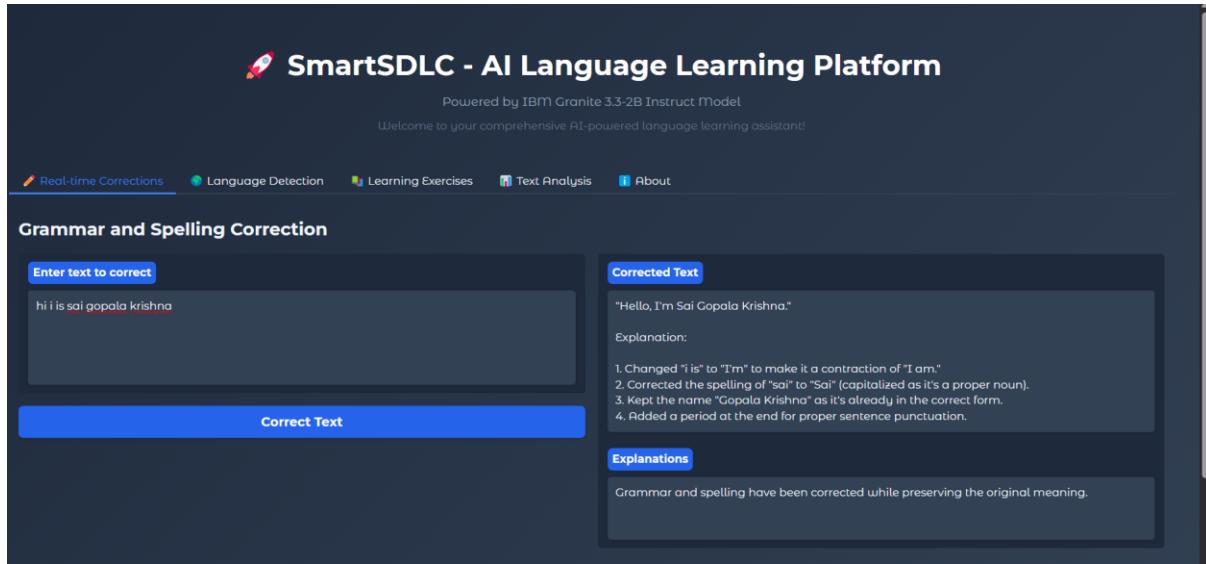
- Only load variables at runtime.

## 7. User Interface

The app features a sidebar for navigation and module selection. Each module accepts different input types and shows results using `st.code`, `st.text\_area`, or `st.chat\_input`.

Modules:

- ❑ **User Input** → Gradio interface captures text/language selection
- ❑ **Language Detection** → LangID identifies input language with confidence
- ❑ **AI Processing** → IBM Granite processes text for corrections/exercises
- ❑ **Analytics Generation** → System calculates metrics and generates visuals
- ❑ **Response Delivery** → Results displayed with explanations and charts
- ❑ **Progress Tracking** → User competency updated and visualized



## 8. Testing

SmartSDLC includes several test mechanisms: -

- Unit Testing: For prompt creation, output cleaning functions.

Integration Testing: Streamlit frontend with IBM Watsonx API.

- 🐛 Manual Testing: For all 6 features (input validation, output quality).

Tests are either in-code validation or handled by test cases generated using SmartSDLC's own test generator module

## Screenshots:

The screenshot shows the "Grammar and Spelling Correction" section of the platform. A user has entered the text "hi i is sai gopala krishna". The platform has corrected it to "Hello, I'm Sai Gopala Krishna." Below the corrected text, there is an "Explanation" section detailing the changes made:

1. Changed "i is" to "I'm" to make it a contraction of "I am."
2. Corrected the spelling of "sai" to "Sai" (capitalized as it's a proper noun).
3. Kept the name "Gopala Krishna" as it's already in the correct form.
4. Added a period at the end for proper sentence punctuation.

Below the explanation, a note states: "Grammar and spelling have been corrected while preserving the original meaning."

At the bottom of the window, the system tray shows the date as 27-06-2025 and the time as 14:00.

The screenshot shows the "Automatic Language Detection" section. A user has entered the text "Hola, ¿cómo estás? Me llamo María y vivo en Barcelona.". The platform has detected the language as "Spanish" with a "Confidence Level: 100%". A note below states: "The text provided is in Spanish. The confidence level is 100% because the text consists entirely of Spanish vocabulary and grammar structures, with no elements that suggest a different language."

At the bottom of the window, the system tray shows the date as 27-06-2025 and the time as 14:01.

The screenshot shows the "Multilingual Learning Exercises" section. The user has selected "Writing Practice" for the "Exercise Type", "English" for the "Language", and "Intermediate" for the "Difficulty/Topic". After clicking the "Generate Exercise" button, a generated exercise prompt appears:

"In the modern era, where waste management is a pressing concern, discuss the significance of recycling in shaping a sustainable future. Highlight at least two key reasons why recycling should be a priority for individuals and communities."

An "Example Answer" is provided:

Recycling is indispensable in our quest for a sustainable future. Firstly, it significantly reduces the amount of waste sent to landfills and incinerators, conserving valuable land resources and minimizing harmful pollution. Secondly, recycling conserves natural resources. By reusing materials like paper, plastic, and metal, we lessen the need for extracting virgin resources, which are often energy-intensive and deplete our planet's finite reserves. These two reasons alone underscore the necessity of recycling for individuals and communities alike.

At the bottom of the window, the system tray shows the date as 27-06-2025 and the time as 14:02.

The screenshot shows a web-based AI-powered language learning platform. At the top, it says "Powered by IBM Granite 3.3-2B Instruct Model" and "Welcome to your comprehensive AI-powered language learning assistant!". Below this is a navigation bar with links: "Real-time Corrections", "Language Detection", "Learning Exercises", "Text Analysis" (which is highlighted in blue), and "About". The main section is titled "Advanced Text Analysis" and contains a text input field with placeholder text: "The quick brown fox jumps over the lazy dog. This sentence contains every letter of the English alphabet. It is commonly used for testing typewriters and computer keyboards. The sentence demonstrates various grammatical structures including articles, adjectives, nouns, verbs, prepositions, and conjunctions.". Below this is a "Analysis Type" dropdown set to "Sentiment Analysis". A large blue button labeled "Analyze Text" is centered. Underneath is a "Analysis Results" section stating: "Sentiment Analysis is a Natural Language Processing (NLP) technique used to determine the emotional tone behind words. It categorizes opinions into positive, negative, and neutral. In the given text, there's no explicit expression of sentiment (positive, negative, or neutral). The text is factual, providing information about a sentence and its linguistic properties, with no subjective opinion or emotional undertone. Therefore, the sentiment analysis of this text would be classified as: Neutral." At the bottom of the window, there's a system tray showing weather (30°C, Mostly cloudy), date (27-06-2025), and time (14:03).

The screenshot shows the "About SmartSDLC" page of the AI-powered language learning platform. At the top, it says "Welcome to your comprehensive AI-powered language learning assistant!" and has a navigation bar with links: "Real-time Corrections", "Language Detection", "Learning Exercises", "Text Analysis" (highlighted in blue), and "About". The main content area is titled "About SmartSDLC" and states: "SmartSDLC is an advanced AI-powered language learning platform that leverages the IBM Granite 3.3-2B Instruct model to provide:". Below this is a "Features:" section with a bulleted list: "Real-time Grammar Correction: Instant grammar and spelling fixes with explanations", "Language Detection: Automatic identification of text language", "Interactive Exercises: Customized learning exercises for multiple languages", "Text Analysis: Advanced linguistic analysis including sentiment, tone, and complexity", and "Multilingual Support: Support for major world languages". There's also a "Technology Stack:" section with a bulleted list: "Model: IBM Granite 3.3-2B Instruct", "Framework: Hugging Face Transformers", "Interface: Gradio", and "Platform: Google Colab compatible". A "Usage Tips:" section follows with a bulleted list: "For best results, keep input text under 500 words", "The model works better with clear, complete sentences", and "Try different difficulty levels in exercises to match your skill level". At the bottom, it says "Built with ❤️ using IBM Granite AI". The bottom of the window shows a system tray with weather (30°C, Mostly cloudy), date (27-06-2025), and time (14:04).

## 10. Known Issues

- No persistent user login system
- No database support (all session-based)
- No role-based access or advanced error handling

- IBM Watsonx API has rate limits depending on your cloud plan

## 11. Future Enhancements

- Add persistent database (MongoDB, PostgreSQL)
- Dockerize for CI/CD deployment
- Implement role-based login system
- Extend to support software architecture generation
- Add support for audio-based prompts or file-to-code generation

## 12. Folder Structure

```
SmartSDLC/ |
├── SMART_SDLC.py → Main Streamlit app file
├── .env → Environment file for secrets
├── requirements.txt → Dependencies
├── /data → Optional sample input files
└── /venv → Python virtual environment
```

### Features:

- **Real-time Grammar Correction:** Instant grammar and spelling fixes with explanations

- **Language Detection:** Automatic identification of text language
- **Interactive Exercises:** Customized learning exercises for multiple languages
- **Text Analysis:** Advanced linguistic analysis including sentiment, tone, and complexity
- **Multilingual Support:** Support for major world languages

## **Technology Stack:**

- **Model:** IBM Granite 3.3-2B Instruct
- **Framework:** Hugging Face Transformers
- **Interface:** Gradio
- **Platform:** Google Colab compatible

## **Conclusion**

The SmartSDLC project successfully demonstrates how AI can revolutionize the Software Development Lifecycle by automating critical phases—from requirement analysis to code generation, testing, and documentation. Leveraging IBM Watsonx's Granite model, this platform bridges the gap between natural language intent and executable software solutions, significantly reducing manual effort and accelerating development cycles.