Automated Book Publication – Project Documentation

Name: Bhavashya Chandra Yadiki
Project: Agentic AI Application – Automated Book Publication

1.Objective

The purpose of this project was to design and build an **agentic AI pipeline** that automates the entire process of book content preparation. The system can:

- Automatically scrape book chapters from the web.
- Use AI to process and rewrite text with improved readability.
- Allow a **human-in-the-loop** step for review and approval.
- Provide a **FastAPI-based interface** to control each stage of the pipeline.
- Store **versioned records** of the processed text in a database for future retrieval.

2.Environment Setup

Commands Executed:

```
bash
# Create and activate a virtual environment
python -m venv venv
venv\Scripts\activate

# Install required packages
pip install playwright requests beautifulsoup4 openai chromadb fastapi
uvicorn pydantic pyttsx3 SpeechRecognition

# Upgrade pip
python -m pip install --upgrade pip

# Install Playwright browser dependencies
python -m playwright install
```

Result: Environment successfully set up. All dependencies installed, and the project was ready to run.

3. Web Scraping and Screenshots

Command Used:

```
python scrape_and_screenshot.py
```

Process:

- Opened a browser session using Playwright.
- Navigated to the target page: *https://en.wikisource.org/wiki/The_Gates_of_Morning/Book_1/Chapter_1*
- Extracted the text content.
- Captured a screenshot of the page for reference.

Files Generated:

- `chapter1.txt` → Extracted text.
- `chapter1.png` → Screenshot of the webpage.

4. AI Writer & Reviewer

Command Used:

```
python ai_writer_reviewer.py
```

Process:

- Loaded the scraped text.
- Used an AI model to "spin" and rewrite the content while keeping meaning intact.
- Performed an AI-based review to refine grammar and flow.

Output:

- AI-generated improved text saved as `chapter1_spun.txt`.

5.Human-in-the-Loop Review

Command Used:

```
python human_feedback_loop.py
```

Process:

- The script displayed AI-written text for manual review.
- User was able to accept or edit the content before final approval.
- Ensured **human oversight** in the automated pipeline.

Output:

- Final approved version saved as `chapter1_final.txt`.

6.FastAPI Endpoints

Command Used:

```
uvicorn agentic_api:app --reload
```

Process:

- Launched a FastAPI server locally at: **http://127.0.0.1:8000/docs**
- Provided an interactive API interface for triggering:
  - `/scrape` → Start scraping
  - `/ai_writer` → Run AI writer
  - `/human_review` → Launch review
  - `/download_final` → Download processed file
  - `/voice_command` → Control pipeline via voice
  - `/rl_feedback` → Reinforcement learning-based feedback loop

Server started successfully and endpoints were tested via Swagger UI.

7.Database Versioning

Command Used:

```
python -c "import db_versioning as db;
text=open('chapter1_final.txt','r',encoding='utf-8').read();
db.save_version('v1', text);
print('Version v1 saved in DB')"
```

Process:

- Saved the final version of the chapter text into the database with a **version tag (v1)**.
- This allows rollback or retrieval of past versions in future updates.

Output:

```
Version v1 saved in DB
```

8.roject Structure

```
automated_book_publication/
│
├── scrape_and_screenshot.py
├── ai_writer_reviewer.py
├── human_feedback_loop.py
├── agentic_api.py
├── db_versioning.py
├── chapter1.txt
├── chapter1.png
├── chapter1_spun.txt
├── chapter1_final.txt
├── requirements.txt
├── DOCUMENTATION.md
├── README.md
└── screenshots/
```

9.Final Deliverables

- **chapter1.txt** – Scraped content
- **chapter1.png** – Screenshot of chapter page
- **chapter1_spun.txt** – AI-processed draft
- **chapter1_final.txt** – Final human-approved version
- **FastAPI interface** with 6 working endpoints
- **Database entry** storing version `v1` of text
- **Proof screenshots** of all executed steps

Screenshots Attached

- CMD showing **successful scraping**
- CMD showing **AI Writer processing**
- CMD showing **Human-in-the-loop approval**
- FastAPI **Swagger UI** with all endpoints
- CMD showing **successful database versioning**

Screenshots are saved in the `screenshots/` folder as proof of execution.

Conclusion

This project successfully demonstrates an **end-to-end automated book publication pipeline** powered by Agentic AI. It can:

- Fetch book chapters online.
- Rewrite and refine text automatically using AI.
- Involve human reviewers for quality control.
- Expose all steps as API endpoints.
- Save versioned text for traceability.

The system is modular, scalable, and can be extended for multi-chapter books, multiple AI models, or automated publishing workflows in the future.