Automated Book Publication Project – Documentation

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

1. Objective

The goal of this task was to build an agentic AI pipeline that automates book content processing. The system is designed to:

- Scrape book chapters from the web.
- Process and rewrite text using AI.
- Allow human review and manual approval.
- Provide a FastAPI interface to trigger each step.
- Store version history of processed text for future retrieval.

2.Environment Setup

Commands used:

Create and activate virtual environment

python -m venv venv

venv\Scripts\activate

Install dependencies

pip install playwright requests beautifulsoup4 openai chromadb fastapi uvicorn pydantic pyttsx3 SpeechRecognition

Upgrade pip

python -m pip install --upgrade pip

Install Playwright browsers

python -m playwright install

Output: Environment successfully set up and ready to run scripts.

3.Phase 1 – Web Scraping

Command:

```
bash
CopyEdit
python scrape_and_screenshot.py
```

Output:

```
arduino
CopyEdit
Starting scraping process...
Launching browser...
Navigating to:
https://en.wikisource.org/wiki/The_Gates_of_Morning/Book_1/Chapter_1
Extracting content...
Taking screenshot...
Scraping complete. Text and screenshot saved.
```

Files generated:

- `chapter1.txt`
- `chapter1.png`

4.Phase 2 – AI Writer & Reviewer

Command:

```
bash
CopyEdit
python ai_writer_reviewer.py
```

Output:

```
arduino
CopyEdit
Starting AI Writer...
Original text length: 9499
Spinning text...
AI Writing and Review complete. File saved as chapter1_spun.txt
```

File generated: `chapter1_spun.txt`

5.Phase 3 – Human-in-the-Loop Review

Command:

```
bash
CopyEdit
python human_feedback_loop.py
```

Output:

```
pgsql
CopyEdit
Human-in-the-Loop Review System
Final approved version saved as chapter1_final.txt
```

File generated: `chapter1_final.txt`

6.Phase 4 – FastAPI API Endpoints

Command:

```
bash
CopyEdit
uvicorn agentic_api:app --reload
```

Output:

```
pgsql
CopyEdit
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```

Accessed the FastAPI documentation at:
http://127.0.0.1:8000/docs

Available Endpoints:

- `/scrape` → Scrape Chapter
- `/ai_writer` → AI Writer
- `/human_review` → Human Review
- `/download_final` → Download Final Text
- `/voice_command` → Voice Command Control
- `/rl_feedback` → RL Feedback Route

7.Phase 5 – Database Versioning

Command:

```
bash
CopyEdit
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')"
```

Output:

```
pgsql
CopyEdit
Version v1 saved in DB
```

Database now contains stored version **v1** of the final text.

8.Project 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 the chapter page
- `chapter1_spun.txt` – AI processed text
- `chapter1_final.txt` – Final approved text
- FastAPI server with 6 working endpoints
- Database with saved version `v1`
- Screenshots folder for proof of execution

10.Screenshots Attached

- CMD showing successful scraping output
- CMD showing AI Writer output
- CMD showing Human-in-the-Loop output
- FastAPI UI (showing all endpoints)
- Database versioning success message

Screenshots are available in the **`screenshots/` folder**.

Conclusion

All phases of the Automated Book Publication project were successfully completed.
The pipeline works end-to-end, providing a fully functional agentic AI solution that scrapes, rewrites, allows human editing, serves via API, and maintains versioned text in the database.