Princess Sumaya University for Technology
Department of Data Science
Autumn Semester 2025/2026

ChronoLearn

A Platform for Learning Jordanian & Palestinian History

Prepared By:
Mohammad ALADDASI 20220630
Shahd Abu Hijleh 20210504

Supervised By:
Dr Omar Alqawasmeh

Project Submitted in partial fulfilment for the degree of Bachelor of Science in Data Science

Autumn - 2025/2026

Declaration of Originality

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university's regulations and is subject to legal actions.

Names and Signatures of team members:

Mohammad O. ALADDASI 20220630

Shahd Abu Hijleh 20210504

Acknowledgments

We would like to sincerely thank Dr Omar Alqawasmeh for his guidance, thoughtful feedback, and continuous support throughout this graduation project. His insights and encouragement contributed significantly to shaping the direction and quality of our work.

We are also heavily grateful to the faculty members of the Data Science and Artificial Intelligence Department at Princess Sumaya University for Technology (PSUT) for providing a supportive academic environment and the educational background necessary to complete this project.

We also would like to express our appreciation to our families for their patience, understanding, and constant encouragement. This gave us the motivation to carry the project through to completion.

Summary

ChronoLearn is an AI-powered educational platform designed to improve the way history is accessed, explored, and understood through direct visualisation. The project is motivated by the limitations of traditional history learning techniques, which often rely on distributed, text-heavy resources that lack contextualisation, interactivity, and representation of relationships between historical concepts.

This project identifies several challenges in digital history education, including inefficient information retrieval from diverse sources, difficulty in capturing the complex dependencies that naturally exist in historical narratives, and the limited availability of advanced educational tools for Arabic historical content. These challenges are further intensified by the linguistic complexity of the Arabic language and the scarcity of specialised NLP resources tailored to historical texts.

ChronoLearn addresses these challenges by designing a user-centric system that integrates Knowledge Graphs (KGs) with Large Language Models (LLMs). Through an ETL-based NLP pipeline, historical documents from multiple formats and sources are processed and transformed into a structured KG that represents entities and relationships. This structured representation serves as a grounding layer for LLM-based semantic querying and content generation, enabling accurate, explainable, and context-aware exploration of historical information.

A key contribution of this project is its focus on Arabic historical narratives, particularly those related to Jordan and Palestine, filling a notable gap in existing digital learning platforms. The system supports interactive outputs such as KG visualisations, and narrative story tale explanations, offering learners multiple intuitive ways to engage with historical content.

The main objective of ChronoLearn is to make historical knowledge more structured and engaging. ChronoLearn targets students, educators, and researchers. Overall, this project demonstrates the potential of combining history knowledge representation using generative AI models, motivating students and scholars for education.

List of Abbreviations

AI: Artificial Intelligence

API: Application Programming Interface

CAMeL: CAMeL Arabic NLP Toolkit

CSV: Comma-Separated Values

DB: Database

EHR: Electronic Health Record

ER: Entity–Relationship

ETL: Extract, Transform, and Load

FAISS: Facebook AI Similarity Search

GP1: Graduation Project Phase 1

GP2: Graduation Project Phase 2

GRAG: Graph Retrieval–Retrieval-Augmented Generation

GUI: Graphical User Interface

HTML: HyperText Markup Language

JSON: JavaScript Object Notation

KG: Knowledge Graph

LAM: Library–Archive–Museum

LLM: Large Language Model

MCQ: Multiple Choice Questions

NED: Named Entity Disambiguation

NEL: Named Entity Linking

NER: Named Entity Recognition

NLP: Natural Language Processing

OCR: Optical Character Recognition

PDF: Portable Document Format

Q&A: Questions and Answers

RAG: Retrieval-Augmented Generation

RDF: Resource Description Framework

RE: Relation Extraction

SPARQL: SPARQL Protocol and RDF Query Language

SQL: Structured Query Language

SRL: Semantic Role Labelling

SQA: Software Quality Assurance

T-Box: Terminology Box

TTL: Terse Triple Language

UI: User Interface

UX: User Experience

XAI: Explainable Artificial Intelligence

Table of Contents

Table of Figures

Table of Tables

Chapter 1

Introduction

1.1 Overview

1.1.1 History as an Academic Discipline

History, as a field of study, seeks to understand how people, societies, and cultures have evolved over time. Traditionally, history education has depended on textbooks, lectures, and memorisation, where students focus on recalling facts, names, and dates. While this approach preserves factual knowledge, it often limits engagement and active learning, making learning history feel static rather than dynamic or exploratory. 95% of students reported higher participation with interactive teaching vs traditional lectures, 81.7% said it increased their motivation. [1]

In its nature, history is data-oriented and time-focused, structured around interconnected frameworks and chronological connections. Every event is part of a larger web that links ideas, people, and places across different eras. However, the way history is often taught does not reflect this interrelationship. The absence of strong storytelling elements or visual and interactive representation results in content that students are expected to memorise rather than truly understand or connect with. In other words, history depends on memory, but it rarely becomes memorable.

Another key issue is the field's resistance to technological integration. Compared to other academic fields, history has been slow to adapt to digital tools and computational methods such as **data visualisation, semantic analysis, and KGs**. Many historians remain cautious, worrying that using data-driven approaches might oversimplify cultural complexity. However, this hesitation has widened the gap between historical research and emerging AI-based methods that could enrich it.

Because of this gap, history risks becoming detached from the digital evolution of knowledge. In a world that increasingly values interactivity, visualisation, and rapid access to information, traditional history teaching methods can feel outdated. The lack of engaging, story-based, and visually supported approaches makes it difficult for learners to connect with historical content in ways that feel meaningful, memorable, and relevant to today's world.

1.1.2 Current Technological Developments for History Education

Current history-seekers are using LLMs in the Q&A mode by querying and receiving answers. Although there are recent LLM models embodied as chatbots, these methods face three main problems with history-seekers:

Firstly, it does not refer to the resources by quote, where the AI explainability black box problem is being faced. Black box refers to hidden and not transparent decision-making processes to users without any explicable reasoning of the results.

Secondly, most chatbots generate text based on linguistic patterns, not factual verification. This problem may lead to hallucinations in the outputs as date confusions, chronological inconsistencies, and context errors, which hinder factual accuracy and reduce its reliability for educational purposes.

Thirdly, LLMs representation in chatbots is becoming a repetitive medium, where reading chunks of texts requires a lot of time and effort to receive the information. Chatbots' means of transferring knowledge to the audience is becoming outdated especially with visual learning techniques.

1.2 Problem Statement

This problem is faced by historians, researchers, and mainly young students who intend to study information of the field of history from digital sources. These text-based data come in the form of articles, books, webpages, photographs, interviews, etc. There is a lack of a unified platform that collects all historical and non-historical resources and archives, and enables the search on it. The traditional, general search of popular search engines such as Google, Bing, DuckDuckGo, etc. return very inexact ambiguous results and restricts the ease of precise information retrieval due to the generalisation features and lack of context-referencing. In addition, most of the online results are text-based outcomes not interactive. Even if it is in an interactional medium such as a visual material or a game, it lacks the topic's focus to cover the intended information.

Simultaneously, a focus on the Arabic Language is needed. The idea is designed to target the Arab world educational community and due to the lack of resources in this language. This product is intended to serve the highest number of Arabic-speaking students.

Therefore, the aim of this project is to design a Jordan and Palestine history context-based KG-LLM, which queries from a user-defined collection of historical and non-historical digital resources. It is done by constructing KGs by retrieving data from different specified sources of different format, such as texts from .docx and .pdf files, webpages through web scraping, or APIs to have the most accurate results regarding the topic. An LLM webpage GUI is to be used as a channel with the user to write the query semantically and returns data in various multimedia elements forms on the queried topic. This will provide semantic and human-like queries that facilitates an efficient data search, on which lacks redundant and imprecise results in an interactive manner.

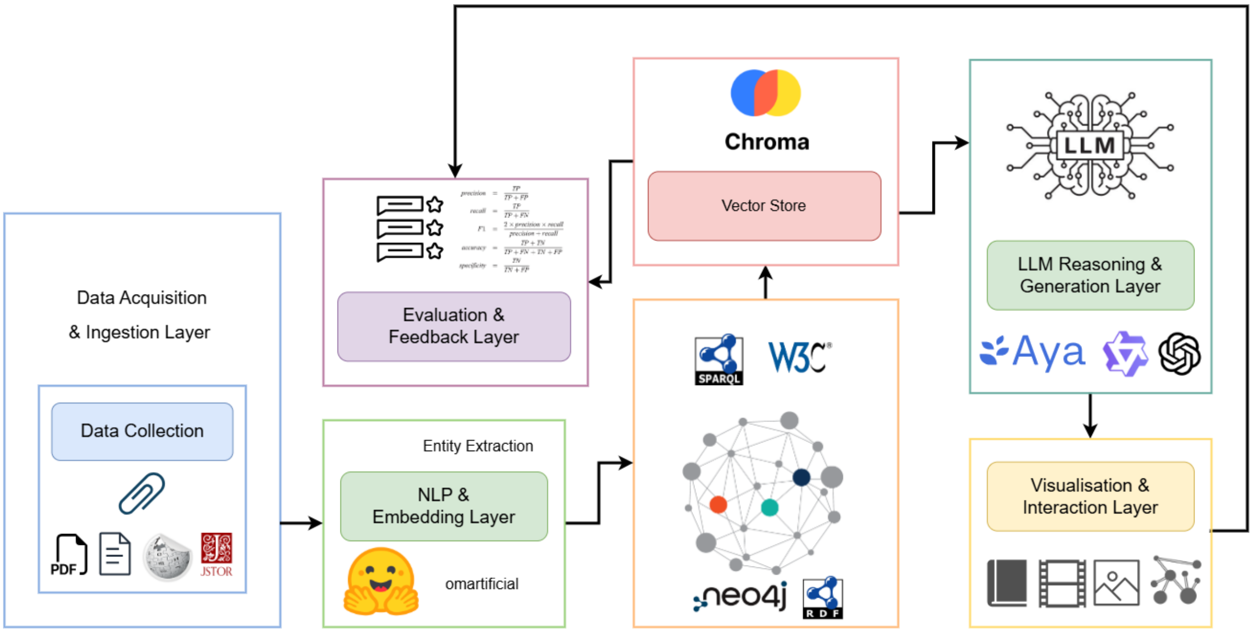

Data Acquisition
& Ingestion Layer
Data Collection
PDF
JSTOR
Entity Extraction
NLP & Embedding Layer
omartificial
$precision = \frac{TP}{TP + FP}$
$recall = \frac{TP}{TP + FN}$
$F1 = \frac{2 \times precision \times recall}{precision + recall}$
$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$
$specificity = \frac{TN}{TN + FP}$
Evaluation & Feedback Layer
Chroma
Vector Store
SPARQL
W3C
neo4j
RDF
LLM
LLM Reasoning & Generation Layer
Aya
Visualisation & Interaction Layer

Figure 1: Initial Knowledge Graph System Design

1.3 Related Work

Our literature review is done systematically and based on answering the following research question:

How can a KG integration with LLM provide an interactive way to represent history discipline in an educational manner compared to traditional text-based representations?

Therefore, according to the initial pipeline of data representation in KG to a final output using an LLM in Figure 1, related works were studied over the three phases of the pipeline methodically. Afterward, it is focused on *three* papers that followed the exact pipeline for a thorough evaluation of the method, contributions, and aims.

1.3.1 Text-to-Knowledge Graphs conversion using NLP methods

Transforming unstructured text into structured KGs is a crucial early step in building intelligent and explainable systems. Through NLP techniques and LLMs, this stage enables narrative or textual data to be represented as interconnected entities and relationships, forming the foundation for reasoning, visualisation, and interactive learning.

Zhang and Soh (2024) [2] developed a framework called Extract-Define-Canonicalise (EDC). It systematically converts text into KGs. The model follows a three-phase approach: extracting potential entities and relations, then defining and aligning them with an existing schema, and finally canonicalising redundant or overlapping nodes. The framework manages larger and more diverse text bodies with minimal manual intervention by incorporating retrieval-based schema linking instead of fixed prompting. This method proved effective in domains where schema scalability and consistency are major challenges. It shows strong accuracy across multiple datasets.

Similarly, **Mohanty (2023)** [3] proposed EduEmbedd, a framework tailored to the educational domain. EduEmbedd primarily focused on embedding KG representations. The model also begins with text analysis through NLP processes such as NER and relation classification. These methods identify both content concepts and pedagogical dependencies; for instance, how one topic supports or precedes another in a curriculum. This framework demonstrates the importance of converting educational texts into graph structures that captures semantic meaning with instructional relationships of educational materials.

In another study, the **Wange et al. (2025)** [4] introduced an interactive, human-in-the-loop approach named the **ChatWeaver** system. Using KG, users and LLMs collaborate to extract, validate, and modify entities and relations in real time through NLP. The system emphasises the value of combining automatic extraction with human verification, enhancing both flexibility and reliability. This approach seems useful for educational and historical data, where contextual precision and interpretability are essential.

1.3.2 Knowledge Graphs-to-LLM employment

KGs are being employed in LLM pipelines by extracting information in semi-structured forms. Such method allows the LLM models to increase results' accuracy by referencing to the sources, making the returned LLM queries results explainable. **Zhang et al. (2024)** [5] designed a framework named KnowGPT: Knowledge Graph-based PromTing that works on solving existing issues with GraphRAG techniques such as large source-search space, high LLM models API cost, and lengthy prompt engineering procedures. The results of this designed model have surpassed the accuracy of GPT-3.5 model by 10.8% and GPT-4 by 3.3% on the In-House test Accuracy (IHtest-Acc), noticeably through the CommonsenseQA dataset mainly, with even higher results on OpenBookQA and MedQA datasets. 5ns.

Akgül et al. (2025) [6] investigated the reasoning capabilities of Large Language Models over temporal KGs by representing time-series data within a dynamic KG framework. Their study was conducted using four datasets, namely ICEWS14, ICEWS18, GDELT, and YAGO. The proposed pipeline includes history sampling for data representation, contrastive learning for entity embedding using contrastive and cross-entropy loss functions, and a test-time filtering step in which LLM-verified entity predictions are returned as responses. The experimental results demonstrated improved performance across both embedding-based and LLM-based evaluations, achieving HITS@k scores for k = 1, 3, and 10 with accuracy improvements ranging from 8% to 35%. However, the framework assumes full observability of historical events, which may limit its applicability in real-world historical scenarios.

Similarly, **Yan, Youfu et al. (2025)** [7] designed a model that is medically focused, which is built on XAI in where KG nodes validate the relationships of the results of an LLM query and even recommends medical advice with information relevancy gradation as supported, relevant and unsure. XNet is tested over ADInt dataset, the evaluation of the model is conducted by gathering weekly feedback from domain experts, which is a limitation since it is qualitative and not quantitatively measured.

1.3.3 Knowledge Graph Visualisation and Interactive deployment of educational LLM end products

An important factor after the processing of the data is by ensuring an interactive output. **Muralidharan et al. (2024)** [8] suggested an interactive yield of the LLM employment using Q&A with multiple-choice questions (MCQs). These MCQs were a successful method as the results were measured by different metrics based on the high values in cosine similarity and the low values in Euclidean distances of the predicted distractor entity embeddings.

1.4 Deep Reviewed Related Works

1.4.1 Paper 1: MedSyn – AI-Driven Medical History Summarisation Using EHR Data [8]

- **Aim and Research Focus:** This study aims to develop an AI-driven system capable of summarising complex Electronic Health Records (EHRs) through RAG on an LLM over a KG dataset. The authors focus on improving clinical information retrieval, to reduce load on clinicians, and enabling efficient question-answering over medical data. The research introduces an integrated pipeline that transforms unstructured EHR content into structured, testable, and interactive context-aware summaries, supporting more informed clinical decision-making.

- **Methodology & Approach:** The methodology adopts a modular, experimental pipeline in this order:
    i. Data Preprocessing: Extracting patient-level (patients, medications, observations) details from a data lake.
    ii. Knowledge Graph Population: Constructing a graph by Neo4j where entity-nodes relationship is built over the patients' data.
    iii. Vector Indexing: Embedding textual representations using BAAI/bge-small-en-v1.5 for similarity-based retrieval.
    iv. RAG Framework Integration: Employing Mixtral-8x7B LLM to generate contextual summaries supported by the knowledge graph and vector store.
    v. Prompt-based Question Answering: Querying predefined clinical questions to generate accurate, explainable, and hallucination-reduced responses.

The entire process transforms raw EHR data into an interpretable KG-based reasoning system capable of answering medical queries and producing more precise clinical summaries.

- **Study Context & Dataset:** The study uses synthetic EHR datasets generated through Synthea, ensuring realistic but privacy-protected clinical data. The dataset includes patient demographics as age and sex, meetings, medications, and observations, all packaged into bundles for preprocessing. This dataset supports the model's summarisation tasks, graph population experiments, and evaluation of question-answering performance.

The use of synthetic data eliminates privacy concerns and aligns with established practices in clinical NLP research while enabling controlled testing environments.

- **Results:** The integrated RAG–LLM–KG architecture significantly enhanced contextual understanding and response accuracy. The pipeline shows an accurate retrieval of patient-specific details when prompts referenced relevant entities. It has improved precision for multi-step clinical reasoning due to the relational structure of the Neo4j knowledge graph. Moreover, summaries were more concise, coherent, and clinically relevant as it is explainable and KG reasoned. A clear reduction in hallucinations when using KG for RAG. Although qualitative in nature, these results highlight the system's strong potential for supporting clinicians with timely, accurate medical insights.

- **Gap Identified (from the authors' perspective):** The authors identify several key gaps:
    1. Lack of Real Clinical Data Testing: Current evaluation relies on synthetic EHRs, limiting generalisation to real-world hospital data.
    2. Absence of Real-Time or Streaming Data: Many clinical workflows require immediate updates, which the current system does not yet support.
    3. Predefined Question Set Limitations: while useful for structured evaluation, these fixed prompts do not cover the full spectrum of clinician query types.
    4. Limited Quantitative Evaluation: summarisation quality and accuracy were assessed qualitatively; the study lacks expert and quantitative validated benchmarks.
    5. Scalability Constraints: KG adds computational overhead, and broader deployment requires more efficient scaling strategies.

1.4.2 Paper 2: Integrated Application of LLM Model and Knowledge Graph in Medical Text Mining and Knowledge Extraction [9]

- **Aim and Research Focus:** This study aims to develop a new approach to enhance the performance of medical question-answering systems by combining the semantic reasoning capabilities of LLMs with the structured relational understanding offered by KGs. The authors seek to improve both information retrieval and answer reasoning in the medical domain, introducing a research direction that integrates intelligent text comprehension with structured knowledge representation to support medical applications.

- **Methodology & Approach:** The research adopts an experimental methodology, collecting medical question-and-answer data from social platforms and ChatGPT outputs. Several detection and evaluation strategies were implemented, including online similarity analysis, offline statistical difference analysis, adversarial generation analysis, and a fine-tuned LLM-based classifier. The process begins by converting raw medical data into a KG, which is then used to support medical text mining through an LLM. This pipeline effectively demonstrates the full transformation of textual medical information into an intelligent reasoning system that can process, analyse, and respond to medical queries.

- **Study Context & Dataset:** The dataset consists of medical question-answer pairs sourced from online social platforms and ChatGPT. These data points were employed to test the proposed detection mechanisms and evaluate the overall system performance. The paper also explores techniques for AI text detection and defence, emphasising the importance of robust model evaluation in medical language tasks.

- **Results**: The integration of LLMs and KGs resulted in more accurate comprehension and extraction of information from medical texts. The findings highlight the potential of this hybrid approach in clinical decision-making, disease prediction, and personalised medical management. Nevertheless, the authors acknowledge that challenges remain in ensuring data diversity and quality assurance, which may influence the scalability and generalisability of the approach

- **Gap Identified (from the author's perspective):** From the authors' perspective, the key gap lies in the lack of integration frameworks that effectively combine LLMs and KGs for domain-specific medical reasoning. Existing systems either rely on LLMs for textual understanding or on KGs for structured relationships but rarely utilise in a unified pipeline. The study highlights the need for more diverse and high-quality datasets, transparent model interpretability, and scalable integration methods that can support real-world medical applications beyond controlled experiments. Addressing these gaps would advance intelligent medical information systems toward more reliable and explainable AI-based healthcare solutions.

1.4.3 Paper 3: The Application of Constructing Knowledge Graph of Oral Historical Archives Resources Based on LLM-RAG [4]

- **Aim and Research Focus**: This study aims to develop a comprehensive method for transforming oral historical archives into structured, machine-readable knowledge using LLM-RAG techniques and semantic KG construction. The main research focus is the challenges of unstructured audio historical materials, such as interviews, manuscripts, and multimedia artifacts, through automated node-edge extractions and querying through LLM.

- **Methodology & Approach:** The research adopts a multi-stage technique. It is based on LLM-RAG for knowledge extraction and a semantic association model for KG construction. The workflow consists of data preprocessing, text segmentation, embedding generation with fine-tuned Chinese language models, vector storage (FAISS), and retrieval through agents integrated with LangChain. Extracted text chunks are then passed through manual human correction to enhance accuracy. Semantic association methods link entities such as persons, events, documents, and spaces to form a structured KG. The KG supports different applications such as "narrative graphs, event-correlation maps, impression clouds, and character-relationship graphs".

- **Study Context & Dataset:** The dataset contains science and art oral historical archival resources from the T.D. Lee Library at Shanghai Jiao Tong University. Historical materials include books, manuscripts, photos, audio and video interviews, letters, and associated artifacts. These resources were catalogued as they are part of the LAM (Library–Archive–Museum) sources and certain labels were added for referencing as geographic, institutional, and authority name datasets.

- **Results:** Through the integration of LLM-RAG, entity detection showed a significant improvement in extracting, organising, and visualising knowledge from oral historical archives. The system successfully generated narrative timelines, event-correlation networks, impression clouds based on sentiment analysis, and objective character-relationship graphs. However, the authors mention that the hybrid LLM-RAG approach increases accuracy and scalability, but it challenges in handling ambiguous Chinese expressions, nested entities, and science or art specific terminology, highlighting the continued need for human-in-the-loop validation and yet the need for more fine-tuning for a whole robust pipeline.

- **Gap Identified (from the author's perspective):** The authors identify several weaknesses in current research and experiment. For instance, there is a lack of a fully automated framework for constructing the KG from oral historical archives, which require manual processing and remain dispersed. It is also noted that there is a limited ability of LLMs in accurately extracting complex or domain-specific Chinese information without fine-tuning or human correction. Moreover, the KG had weak interoperability across archives, libraries, and museum formats, which restricts multi-dimensional resource aggregation. Thus, the authors suggest the need for using refined models, better Chinese language handling, expanded datasets, and standardised frameworks to support fully automated, interpretable, and scalable KG construction.

1.5 Contribution

This research and project present multiple key contributions, methodological innovation, and application-specific relevance.

1.5.1 Novelty of the Idea

The suggested system design focuses on *Jordanian* and *Palestinian* historical narratives through an *Arabic-language,* KG-LLM educational pipeline. Currently, the existing GRAG-LLM based projects primarily target a global English-speaking audience using Western trained data. Therefore, this project addresses regional history and local narrative viewpoints that are underrepresented. The project further advances post-LLM educational systems by offering an *interactive*, multimodal output experience, including KG visualisation, storyboards, and story tale answers.

1.5.2 Target Audience

The system is designed to serve *students, academics, and heritage researchers* seeking structured access to Jordanian and Palestinian history. Its personalised search capabilities support learners conducting focused research, receiving explainable and context-aware outputs. These reasoned results allow educators and researchers to verify sources and trace conceptual linkages. The pipeline is both a learning tool and an interactional research platform. At the same time, its interactivity changes the traditional teaching of history disciplines by the final products, which can lead to expanding the audience to the all the public interested in discovering the history discipline.

1.5.3 Novelty in the Choice of Model

The project introduces state-of-the-art GRAG-LLM hybrid architectures, drawing inspiration from emerging systems such as ChatWeaver [4], EDC [5], and EduEmbedd [3], employing Arabic in language and narrative processing and history-focused entity extraction. The choice of model emphasises semantic depth, explainability, and pedagogical grounding, tailored to the structure of historical texts rather than general-purpose corpora.

1.5.4 Novelty in Pipeline Structure

The system extends existing GRAG-LLM pipelines by integrating:

- Personalised input selection, where users define the web sources, materials catalogues, and digitised archives used for knowledge extraction.
- A hybrid entity-relation extraction module in Arabic historical semantics.
- Interactive post-processing outputs, enabling exploration of KG nodes, attributes, textual (story tale) and visual story-based media.

This produces a dynamic, learner centric pipeline rather than a static extraction model.

Objectives that this study fulfils:

1. Personalisation

   Choice of websites, digitalised catalogues, resources, etc. (inputs) is done by the user to conduct the research queries on. Besides, focused and customised interactive results are found on the browsing history of the user (outputs).

2. Context-Aware Results

Ability of recommending related results based on the user's inputs to exclude any anomalies of the semantically defined data represented in triples.

3. Explainability

Employs XAI in interactive results, such that the recommendations are contextualised and can be traced back to the sources with the specified parameters, in which accuracy of outcomes is obtained in a higher level and resources and citation are navigated.

1.6 Document Outline

This report is structured to present the conceptual foundation, design, and implementation of the proposed history-based KG-LLM system (ChronoLearn) in a clear and systematic manner.

Table 1: Document Outline Per Chapter

Chapter / Section	**Description**
Chapter 1 – Introduction	Presents the project background, including history as an academic discipline and current technological developments in history education. It defines the problem statement, reviews related work for different pipeline stages, highlights the project's contributions and novelty, and concludes with the document outline.
Chapter 2 – Project Plan	Describes the overall project plan, including project deliverables, detailed project tasks, roles and responsibilities, and a risk assessment identifying potential risks and mitigation strategies.
Chapter 3 – Requirements Specification	Specifies the system requirements by identifying stakeholders and target audiences, defining platform requirements, functional and non-functional requirements, and other relevant constraints.
Chapter 4 – System Design	Details the system design, covering the architectural design, logical model design, and physical model design, and explains how system components are structured and interact.
Chapter 5 – Data Preprocessing	Explains the data preprocessing stage, including data collection and description, data profiling and engineering, feature engineering, and data loading procedures used to prepare the data for subsequent stages.

Chapter 2

Project Plan

2.1 Project Deliverables

The project aims to implement a complete AI-powered history-based educational platform that transforms multi-sourced textual data into interactive end products. The interface primarily must be user-friendly and a convenient solution to learn history.

The system must read the Arabic text documents, with an interpretation in terms of Jordan and Palestine's history. The backend should follow an ETL-based NLP pipeline information pre-processing and follow the KG construction techniques using LLM API calls. The KG exports its nodes into stories as a source to feed the LLM for the end-user interactive products such as a text story tale, storyboard, including a knowledge graph.

The components of the project are as follows: *Website (frontend)*, *GRAG-LLM System (backend)*, the user-provided datasets and the documentations in its forms, are described below:

1. **Website (frontend):** The website serves as the educational platform's primary frontend, presented through a web interface. It enables users to access their accounts, attach documents, visualise the information in various forms, and search the KG. The website should be flexible and easy to navigate for users of different experience levels from young students to older teachers. The tool planned for use is Lovable for the development of the UI integration in a TypeScript file format.
2. **GRAG-LLM system (backend):** The pipeline of the project is based on the GRAG-LLM system, which is responsible for the breakdown of the historical documents, and use the state-of-the-art technologies for text comprehension with LLM, NER extraction to build the KG, designing different visual products such as the interactive KG, storyboard, and text-story.
3. **Dataset:** Historical documents: The data used must be a sample data to design and build the system over. The data is collected from different Arabic typed academic historical-topic documents, online articles, and textbooks. Whereas in the published system, the data is user-based documents uploaded on the platform for the processing part.
4. **Documentation:** Three documents are to be produced by the end of the project. A **report** that documents the full procedure of the idea and project development such that analysis, planning, implementation, and evaluation. A **user manual** for the users on the platform (frontend) and how to use it step-by-step explaining the app for the target users. A **pitch presentation** for both technical and non-technical target users. These documents are essential deliverables to ensure the project is clearly communicated, evaluated and guides the target audience.

2.2 Project Tasks

The framework of the project divides into several parallel and/or simultaneous stages composed of multiple consecutive tasks. This methodology allows the most efficient and systematic structure of the project's launching allowing future replication. The documentation of the theoretical design is to ensure reproducibility and traceability of the applied production of the project's pipeline.

ChronoLearn follows an ETL-pipeline of textual data ingestion undergoing NLP, transforming through GRAG-based LLM, then producing a tangible user-friendly product. After the final product is working, an evaluation is planned to verify the results of the pipeline using different metrics and testing methods. Alongside, on-paper documentation of the process is employed to record the academic procedures, obstacles and results of the experiment.

2.2.1 Analysis & Research

Task 1: Review related-work and research

- Review of extensive literature and academic papers related to students' methods of learning Jordanian and Palestinian history.
- Study literature and academic papers employing KG, GRAG, and LLM with existing research on history discipline educational platforms.

Task 2: Market Need Assessment

- Conduct interviews with domain experts, educators and historians, to understand educational hardships and set specifications for the final product.
- Perform comparative study of existing historical KG, digital history platforms, and educational.

Task 3: Data and Dataset Requirements

- Select a sample of qualitative relevant sources for the experiment's application.
- Define input formats (DOCX, PDF, JSON, and HTML).
- Specify expected outputs (Interactive KG, AI-generated stories, visualisations) and performance indicators.

Ethical considerations

The sample data must be extracted from open-source and not-copy-reserved resources to avoid any plagiarism and infringement of copyrights issues.

2.2.2 Design

Below shows the design tasks for the architecture, UI, and DB elements:

Architecture Design

Task 1: High-Level System Architecture Draft

- Design an initial high-level framework that outlines the general pipeline of the data.
- The model includes subsidiary modules such as input data collection, NLP, KG building, embedding layer, KG-to-Text Layer, vector-store, LLM generation layer, and the user interface layer.

Task 2: Detailed Workflow & Pipeline Design:

- Design a detailed step-by-step flowchart based on the high-level design architecture, which explains each layer in detail.
- The model explains the inputs and expected outputs of each layer, and which channels connect them together.

UI Design

Task 1: UI Requirements List & Mapping Interaction

- Identify UI requirements including user interactions and roles.
- Define the users' access to the interface.

Task 2: Design Wireframes

- Illustrate a mock-up to visualise screen layouts, components, menus, and navigation flows.

Task 3: Develop UI Prototype

- Build interface prototype specifying colour schemes, typography, icons, component styling, and user interaction behaviour.
- The prototype represents the final visual blueprint for implementation.

Database Design

Task 1: User Credential Schema Design

- Design the DB schema managing user accounts.

Task 2: ER Diagram & Data Constraints Design

- Create an Entity–Relationship (ER) diagram composed of credentials tables and relationships.

2.2.3 Implementation

Below shows the pipeline execution tasks this includes the coding and programme development phases:

Task 1: Template KG Setup

- **Populate core entity types and entity-relationships relevant to Jordanian & Palestinian history:**
    - Persons: (leaders, historical figures, activists)
    - Events: (wars, treaties, migrations, social movements)
    - Locations: (cities, towns, refugee camps, historical sites)
    - Organisations: (political parties, committees, resistance groups)
    - Documents/ Sources: (archives, newspapers, official letters)
- **Create the KG template schema**:
    - Determine granularity of basic nodes:
        - Include details and determine its depth per each entity.
    - Map the triples of the nodes and edges to validate structure.
    - Ensure the flexibility for mapping the future sources-based extracted entities.

Task 2: Data Collection & Pre-processing (NLP)

- **Collect raw data:**
    - Scrape and extract Arabic-based texts from open access archives and catalogues.
        - E.g. Wikipedia, JSTOR, history books of Jordan and Palestine.
- **Preprocess text:**
    - Convert PDF/HTML to plain text.
    - Normalise Arabic text (remove supplementary diacritics as harakat, unify characters, etc.).
    - Tokenisation, stemming, lemmatisation, and sentence splitting.
- **Data cleaning:**
    - Remove irrelevant sections and stop words.
    - Handle OCR errors or mis-encoded characters; if possible.
    - Store data in organised directories with metadata.

Task 3: NER Extraction

- **Semantic Role Labelling (SRL):**
    - Apply NER and dependency parsing using LLM through API calls.
    - Identify entities as persons, locations, dates, events, and organisations, and relationships as “participated in,” “founded,” “located in”.
    - Map relationships in triples: (subject, predicate, object).

Task 4: KG Modelling & Building

- **KG Construction:**
    - Import the KG schema of induced KG schema and mapped triples.
    - Populate KG with extracted triples.
    - Ensure consistency (e.g., canonical names for entities).
    - Store KG in a DB (Neo4j, RDF, or JSON-LD format).

    - Implement versioning for iterative updates.
    - Map historical sources per each node for reasoning referencing.
    - Ensure consistency of entity names (canonicalisation).

Task 5. LLM Story Generation from KG

- **Vector-store Text Data Generation**
    - Convert KG nodes, edges, and subgraphs into historical stories using an LLM.
    - Build story tales over a chronological order including all the persons, locations, and events nodes.
    - Cite each statement by hyperlinking it to the resource from its corresponding extracted node.
- **Generation Accuracy assurance**
    - Restrict model hallucinations by optimising the prompts.
    - Refer to KG facts as the grounding source.

Task 6. Vector-Store Building and Embedding Pipeline

- **Story Embedding and Vector-store Validation**
    - Choose an Arabic-compatible embedding model.
    - Create and initialise the vector-store collections.
    - Generate dense vector representation of the stories' texts.
    - Embedding metadata and resources.
- **Retrieve Data from the Vector-store**
    - Execute similarity search.
    - Ensure correct retrieval by checking retrieval coefficients.
    - Setup versioned updates of vector-store by regenerating and re-embedding when KG expands.

Task 7: Project Interactive Interface Development

- **UI Implementation**
    - Develop a web-based dashboard that serves as the users' access point for all the project outputs.
    - Organise the interface into modules:
        - Knowledge Graph Exploration
        - Text-based Story Tale
        - Storyboard Viewer
    - Ensure responsive layout and multilingual support (Arabic/English).

2.2.4 Testing & Evaluation

Below shows the projects' assessment tasks this includes, the programme execution verification, SQA, and KG and AI-generated products validation phases:

Task 1: Programme Testing

- **Data Integrity Testing**
    - Check that triples are correctly constructed with entities and relations following the existing schema.
    - Ensure no duplicates or inconsistent entries appear in the KG.
- **System Performance Evaluation**
    - Test vector-store retrieval and generation time from LLM.
- **User Testing**
    - Evaluate ease of use for students and historians through task-based testing.
    - Request qualitative expert feedback and appraisal on accuracy, clarity, and educational value.
- **Software Quality Assurance (SQA)**
    - Verify that all interface components such as KG visualisation, storyboard, story tale text output function as intended.
    - Consider evaluating the GRAG-LLM results by the following *potential* metrics: BERTScore, Recall@K, Precision@k, HITS@k, Temporal Accuracy (TA), Rouge-N and Rouge-L measures, etc.

Task 2: Bug Fixing & Iterative Improvement

- Identify issues from different testing techniques, prioritise them, and apply debugs.
- Enhance interaction flow and performance based on expert and user feedback.

Task 3: Validation

- Review manually samples of KG entities, relations, and generated stories for correctness.
- Adjust rules, prompts, or extraction logic when recurring errors appear.

2.2.5 Documentation

Concurrent with the operation of the previous tasks, documentation is established and continuously updated. It includes recording of design decisions, methodologies, data processing steps, system architecture, evaluation results, and revisions made throughout both phases of researching and practical development. Simultaneously, manuals and tutorials are designed for the optimal user experience. These documents ensure a final traceable state-of-the-art method recording report.

Table 2: Summary of Tasks, Subtasks - if available, and Time Estimation Per Task.

	Task	**Subtask - if available**	**Time Duration (weeks)**
Analysis & Research			
1.	**Review related work and research**	N/A	2
2.	**Market Need Assessment**	N/A	1
3.	**Data and Dataset Requirements**	N/A	1
Analysis & Research - Architecture Design			
4.	**High-Level System Architecture Draft**	N/A	1
5.	**Detailed Workflow & Pipeline Design**	N/A	1.5
Analysis & Research - UI Design			
6.	**UI Requirements List & Mapping Interaction**	N/A	1
7.	**Design Wireframes**	N/A	1
8.	**Develop UI Prototype**	N/A	1
Analysis & Research - DB Design			
9.	**User Credential Schema Design**	N/A	0.5
10.	**RE Diagram & Data Constraints Design**	N/A	1
Implementation			
11.	**Template Knowledge Graph Setup**	Populate core entity types and	1.5

		entity-relationships relevant to Jordanian & Palestinian history:	
		Create the KG template schema	
12.	**Data Collection & Pre-processing (NLP)**	Collect raw data	1
		Preprocess text	1
		Data Cleaning	1
13.	**Entity-and-Relation Extraction**	Semantic Role Labelling (SRL)	
14.	**Knowledge Graph Modelling & Building**	KG Construction	3
15.	**LLM Story Generation from Knowledge Graph**	Vector-store Text Data Generation	1
		Generation Accuracy Assurance	1
16.	**Vector-Store Building and Embedding Pipeline**	Story Embedding and Vector-store Validation	1.5
		Retrieve Data from the Vector-store	1
17.	**Project Interactive Interface Development**	Design UI	concurrent
		Implement Search & Query Functions	2
Testing & Evaluation			
18.	**Programme Testing**	Data Integrity Testing	0.5
		System Performance Evaluation	0.5

		Software Quality Assurance (SQA)	1
		User Testing	1
19.	**Bug Fixing & Iterative Improvement**	N/A	1
20.	**Validation**	N/A	0.5
Sum of Weeks for concurrent 20 Tasks (approximately)			32

The following is a dynamic Gantt Chart that demonstrates the timeline of the project. The chart is uploaded on the GitHub page in mermaid.js:

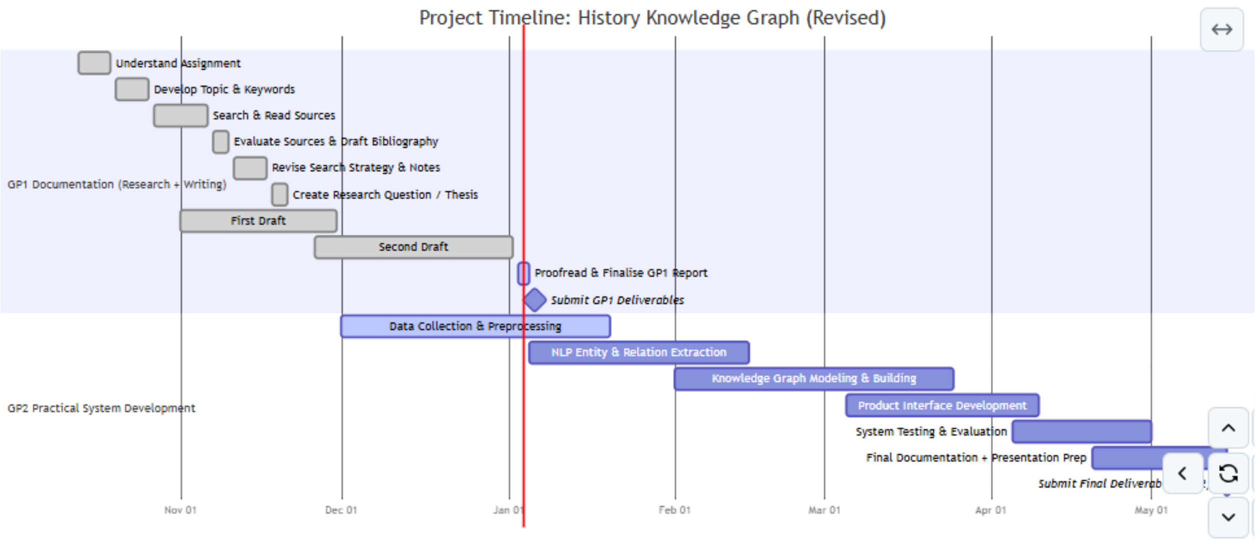


Figure 2: Gantt Chart of the Project Timeline over GP1 and GP2 phases

2.3 Roles and Responsibilities

The project is undertaken as a pair-work collaboration. With cooperative research, an equitable task distribution and mutual accountability was done. Responsibilities were assigned simultaneously to both members, to ensure that both contributed effectively to project's objectives achievement. Moreover, despite the presence of individually managed components, all stages of development are jointly discussed, reviewed, and refined. This is to maintain a methodological consistency and alignment of the project's overall goals.

2.4 Risk Assessment

Table 3: List of Risk Assessment per Task

Task Section	***Task***	***Identified Risk***	***Probability***	***Impact***	***Mitigation Strategy***
Analysis & Research	Review related work	Limited availability of Arabic research on historical KG/LLM systems	Medium	Medium	Expand search scope (Arabic NLP, digital learning) and consult regional experts
	Market need assessment	Difficulty coordinating interviews with teachers/historians	Medium	Low	Use online forms or email-based questionnaires
	Dataset requirements	Limited availability of open Arabic historical resources	High	Medium	Use public archives and build a small, curated dataset
Architecture Design	System architecture draft	Architecture may become too complex at early stages	Medium	High	Start with a simplified structure and expand gradually
	Workflow & pipeline design	Workflow steps may not connect smoothly	Medium	Medium	Conduct regular reviews of workflow diagrams
UI Design	Requirements mapping	Users may express unclear or incomplete needs	Medium	Low	Provide examples and refine requirements iteratively

	Wireframes	Early sketches may not reflect real user navigation	Low	Low	Test wireframes with a small user group
	UI prototype	RTL (Arabic) layout and alignment issues	Medium	Medium	Use UI libraries that support RTL rendering
DB Design	User credential schema	Risk of insecure or weak authentication flow	Low	High	Apply trusted authentication libraries and hashing standards
KG Setup	KG template setup	KG may not accurately represent historical relationships	Medium	High	Review structure with history experts and refine as needed
	Entity duplication	Multiple Arabic spellings causing duplicate nodes	High	Medium	Apply canonicalisation rules and merge similar names
NLP Preprocessing	OCR for Arabic text	OCR tools may produce significant errors	High	Medium	Use Arabic-optimised OCR tools; manually correct key sections
	Tokenisation & stemming	Arabic NLP tools may mis-handle classical/historical text	Medium	Medium	Use Arabic-specific tools (Farasa, CAMeL)
	Long documents	Text exceeds LLM input limits	High	Low	Chunk or summarise text before processing

Entity & RE	Entity extraction	LLM may misidentify people, places, or events	Medium	High	Use text-grounded prompts and recheck uncertain cases
	Relation extraction	Incorrect or incomplete historical relationships	Medium	High	Add confidence scoring; manually review low-confidence triples
KG Construction	Duplicate or messy KG nodes	Inconsistent or inaccurate KG structure	Medium	High	Apply canonicalisation, name standardisation, and merging procedures
	Incorrect triple storage	Triples stored incorrectly, affecting queries	Low	High	Test small samples before full ingestion
Story Generation	Historical inaccuracies in LLM stories	LLM may add information that does not present in KG	Medium	High	Enforce KG grounding and require citations in responses
	Incorrect timeline ordering	Events appear out of sequence	Low	Medium	Extract and sort date information before generation
Vector-Store & Embeddings	Weak Arabic embeddings	Meaning not captured accurately	Medium	High	Use high-quality Arabic embedding models
	Irrelevant retrieval	Retrieved text may not match user query	Medium	Medium	Adjust chunk size and

					similarity thresholds
Interface Development	Slow KG visualisation	Large graphs may load slowly	Medium	Medium	Use lazy loading and node clustering
	Inaccurate search results	System may return unexpected or irrelevant results	Medium	Medium	Improve ranking algorithms and add filtering options
Testing	Wrong triples → wrong stories	Incorrect outputs affecting reliability	Medium	High	Perform manual reviews and automated checks
	Slow system performance	System lag affects user experience	Medium	Medium	Optimise queries and apply caching
	Users struggle with UI	Interface may be confusing for students	Low	Medium	Add tooltips, labels, and clear instructions
Bug Fixing	Fixes causing new issues	New bugs introduced accidentally	Medium	Medium	Use version control and regression testing
	Limited time for fixes	Not all issues may be resolved before deadline	Medium	Medium	Prioritise high-impact and user-visible issues
Validation	Historical inaccuracies remain	Risk of incorrect or misleading information	Medium	High	Perform expert reviews and check multiple sources

	Sensitive political content	LLM may produce inappropriate or biased text	Medium	High	Use safety prompts and manually review sensitive outputs

2.5 Cost Estimation

For this project, the budget is of an estimated cost of 500 JDs for image generation, LLM deployments, processing runtime enhancements. Detailed prices breakdown is determined when implementation phase starts as most of the required costs are in pay-as-you-go pricing system.

2.6 Project Management Tools

Table 4: List of Project Management Tools & Technologies

Component	Tools / Technologies
Task Management & code version control	GitHub
Flowcharts & Visual Diagrams	Draw.io
Word Processer	Microsoft Word
Programming	Python - Jupyter Notebook
Embedding	Omartificial
Knowledge Graph	Neo4j or RDFLib
Vector Store	ChromaDB
Data Pipeline	Pandas, BeautifulSoup, Selenium
LLM Models	Aya Model, Qwen, OSS2b (if feasible)
Evaluation	Precision / Recall, SPARQL Queries, Cypher
Visualisation	Cytoscape.js, Vis Network (Vis.js), Sigma.js, D3.js

Chapter 3

Requirements Specification

3.1 Stakeholders & Target Audience

3.1.1 Stakeholders

Stakeholders are individuals or entities that influence or are affected by the system's design, development, or operation. They are categorised into large-scale and small-scale groups.

Large-Scale Stakeholders:

1. **Educational Institutions (Schools, Universities, Learning Centres):** Institutions can integrate the system into their digital learning platforms to enhance history education through interactive knowledge graphs and AI-based exploration.
2. **Curriculum Designers and Education Ministries:** Ensure the system aligns with official academic standards, maintaining historical accuracy and relevance within educational frameworks. As this segment engage with the system to align AI-driven historical content with national and institutional learning outcomes. They play a regulatory and evaluative role. It guides how the system supports learning objectives and assessments.
3. **Historical Research Centres and Digital Humanities Labs:** Use the system's data representation and semantic linking features to enhance historical databases and improve research accessibility. This contributes to its scholarly credibility and integration in academic research.

Small-Scale Stakeholders:

1. **Teachers and Educators:** Use the system to simplify complex historical topics, design engaging lessons, and provide valuable feedback for system improvement. They also provide critical feedback that informs system refinement and academic alignment. Through their interaction with students, teachers can also collect and relay feedback on learning outcomes, engagement levels, and comprehension difficulties. This student-based input helps refine the system's educational design, ensuring that its content, visualisations, and AI interactions remain aligned with learners' needs and academic objectives.

2. **Historians:** Contribute verified historical data, ensuring factual accuracy, diversity, and inclusiveness in the represented knowledge.

3.1.2 Target Audience

The target audience includes the end users who directly engage with and benefit from the system. They are divided into large-scale and small-scale user groups.

Large-Scale Target Audience:

1. **Educational Institutions:** These institutions integrate the system into their existing learning frameworks to modernise history education. By adopting AI-driven visualisation and interactive storytelling, they can enhance engagement and comprehension among students while ensuring alignment with institutional learning standards.

Small-Scale Target Audience:

1. **Students and Learners:** represent the core users of the system, interacting directly with its tools to explore historical content in more engaging and dynamic ways. The system allows them to navigate historical timelines, understand cause-and-effect relationships, and visualise interconnected events, transforming history learning from memorisation-based study into an interactive experience.
2. **Teachers and Academic Staff:** use the system to enrich their teaching methods by presenting history through interactive visuals, knowledge graphs, and AI-assisted explanations. This enables them to simplify complex historical connections, promote inquiry-based learning, and encourage student participation while saving time in lesson preparation.
3. **Historians and Researchers:** Historians and researchers can use the system as a practical tool to explore and organise historical information in new ways. By combining KG with LLM, the platform helps them connect events, people, and sources more intuitively. It also supports information verification by tracing evidence turning traditional research into a more dynamic, interactive, and insightful process.

3.2 Platform Requirements

The platform must be a client–server architecture. The server runs all the tasks such as the NLP pipeline, ontology reasoning, vector storage, KG DB, and backend APIs. A Linux server such as Ubuntu 20.04+ is required. Python 3.10+ and a backend such as Flask or FastAPI are required. Neo4j/RDF storage and ChromaDB are also required. LLM API access is required. A web server such as Nginx or AWS must be used.

The minimum hardware is a 4-core CPU, 16 GB RAM, 200 GB SSD storage, and 50 Mbps internet. For higher loads, an 8–16 core CPU, 32–64 GB RAM, and a 500 GB NVMe SSD are recommended. A GPU is optional but helps. Backups and monitoring are also recommended.

The client must run in a modern web browser such as Chrome, Edge, or Safari. JavaScript must be enabled. Internet access is required. Any common OS may be used, such as Windows, macOS, or Linux. The minimum hardware for the client is a dual-core CPU, 4 GB RAM, and a 1366×768 display. These are mandatory. Better performance is achieved with 8 GB RAM, a quad-core CPU, and a Full-HD display. GPU acceleration is recommended. These are optional but useful.

3.3 Functional Requirements

Table 5: Detailed Functional Requirements List

Requirement Name	**Type**	**Input**	**Processes**	**Output**	**Main Constraints / Dependencies**
Document Upload	Essential	Historical document or text file	User attaches/uploads their documents and webpages to the designated section	Uploaded file successfully processed for analysis	Limited to specific formats; requires internet and stable upload connection
Document Summarisation	Essential	Large text document	System condenses long input into key ideas, entities, and timelines	Concise summary ready for visualisation or story generation	Requires NLP model availability
Knowledge Graph Generation	Essential	Processed text data	Extracts and links entities, dates, and events to build a structured graph	Interactive knowledge graph with entity relationships	Depends on accurate entity recognition and graph DB connectivity
Interactive Story Generation	Essential	Summarised text or user-selected topic	Generates historical narratives reflecting causal and chronological relationships	AI-produced story text providing historical insight	Relies on LLM accuracy; requires validated training data; must ensure

					factual correctness
Storyboard Visualisation	Essential	Generated story or Knowledge Graph data	Converts historical data into interactive storyboards	Visual display illustrating historical flow and content	Requires stable browser rendering; visualisation library compatibility
Knowledge Graph Query	Essential	Keywords or natural language query	Searches the graph to retrieve relevant nodes and their connections	Retrieved entities and linked relationships displayed visually	Relies on efficient semantic search; depends on current KG dataset
User Account Management	Essential	Registration details, login credentials, previous interactions	Manages identity and enables saving history, sessions, and preferences	Secure user account with retained interaction record	Requires authentication services; encrypted storage; privacy compliance
User Feedback Collection	Recommended	User comments or ratings	Collects qualitative & quantitative feedback for system enhancement	Feedback reports supporting continuous improvements	Requires analytics module; voluntary participation
Source Citation and Traceability	Essential	Generated text or KG data	Traces and references original sources linked to historical facts	List of verified references linked to generated outputs	Needs access to reference database; requires indexing for source–output linking

3.4 Non-Functional Requirements

In addition to the functional requirements necessary for the system to operate correctly, the non-functional requirements listed in Table 7 ensure the overall usability, reliability, performance, and quality of the KG Historical System

Table 6: Detailed Non-Functional Requirements List

Category	Non-Functional Requirement	Example in the KG Historical System
Performance	The system must process NER, RE, KG queries, and document ingestion with the most minimum time, even when handling long historical texts.	The system extracts historical entities (e.g., places, dates, persons) from any provided archival document in under 10 seconds and returns NER, then KG visualisation instantly.
Scalability	The system must scale horizontally to support large datasets, thousands of documents, and complex KG expansions without performance loss, with cost limitation.	When users upload numerous historical documents, the system automatically allocates more compute/graph storage to accommodate new nodes and relations with a limit for the number of files uploaded and character count.
Security & Access Control	All user-uploaded archives, extracted semantic data, and generated KG structures must be securely stored, encrypted, and permission-controlled.	User documents are encrypted at rest, admin and viewer roles are separated, and KG editing rights are restricted to authorised researchers.
User Experience (UX)	The interface should be intuitive, with smooth navigation, interactive graph exploration, and clear visual feedback for each action in a sequential order.	Users can click on KG nodes to view sources, and open narrative summaries without technical expertise.

Uptime / Reliability	The system must maintain a minimum of 99.9% uptime and recover automatically from failures.	The system is deployed on distributed cloud architecture with automatic failover for graph DBs and APIs.
Documentation	The system must include complete technical documentation, architecture diagrams, user guides, and API references.	Documentation includes instructions for dataset ingestion, KG schema definitions, NER pipeline, and API endpoints for external integrations.
Interactivity	The system must support interactive exploration of KG nodes, visual material, and multi-modal historical media.	Users can traverse the KG, compare historical narratives, view linked photos/videos, and overlay temporal relationships in real time.

3.5 Other Requirements

Table 7: Detailed Other Requirements List

Category	Additional Requirement	Example in the KG Historical System
Bias & Prompt Engineering	The system must implement bias-mitigation strategies in all LLM prompts to prevent cultural, political, or ideological distortion.	Prompt templates ensure balanced entity extraction and narrative generation for sensitive historical topics (e.g., Nakba, regional conflicts).
Large-Scale Pipeline	The system must support large-scale, end-to-end processing of historical material with efficient handling of long documents and high-volume datasets.	The pipeline processes hundreds of archival pages, performs OCR, entity linking, clustering, and KG generation using distributed or batch processing.

Online Data Access	The system must be able to retrieve, cross-check, and integrate data from online open source verified libraries such as Wikipedia, DBpedia, digital archives, and open APIs.	When a historical location or figure is ambiguous, the system queries Wikipedia or Wikidata to enrich context and validate extracted entities.
Data Governance & Attribution	The system must maintain complete provenance metadata for all extracted information to ensure transparency and traceability.	Every KG node stores citation details (document source, page number, time of extraction, and external URL -if found-).
Ethical & Cultural Sensitivity	The system must handle sensitive historical narratives responsibly, avoiding misinterpretations or oversimplification.	When conflicting sources are detected, the system highlights uncertainty levels and provides alternative interpretations rather than a single deterministic output.

Chapter 4

System Design

This chapter describes the proposed *Semantic Knowledge Graph Extraction System* from an architectural, logical, and physical perspective. The goal of this chapter is to present how the system components interact, how users engage with the system, and how the underlying processing pipeline transforms uploaded documents into validated RDF-compliant knowledge graphs.

4.1 Architectural Design:

The architectural design provides a high-level view of the system. It illustrates major components including the UI layer, the NLP & NER pipeline, the ontology-driven reasoning layer (T-Box), the knowledge graph storage, the vector storage & LLM reasoning layer, the evaluation & feedback loop, and the visualisation/export components.

These relationships are shown in the diagrams, taken as screenshots in the GitHub repository.

4.1.1 High-Level System Architecture

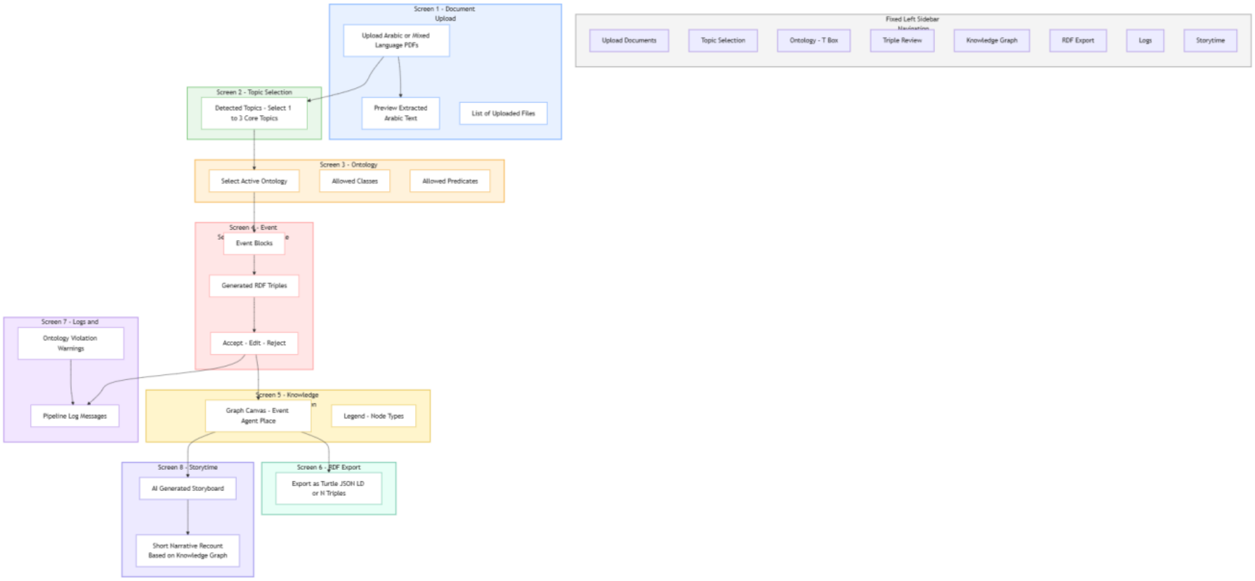


Figure 3: High_Level_Architecture_Diagram.md

This diagram shows the main building blocks of the system and how they communicate. It highlights the role of the LLM reasoning module, Chroma vector store usage, Neo4j / RDF storage, and entity extraction layer together with user-facing components such as the visualisation and interaction layer.

This diagram is important, because it demonstrates that the system contains a human-in-the-loop with model ontology (T-Box) built with it rather than an uncontrolled generative system.

4.1.2 Initial Knowledge Graph System Design

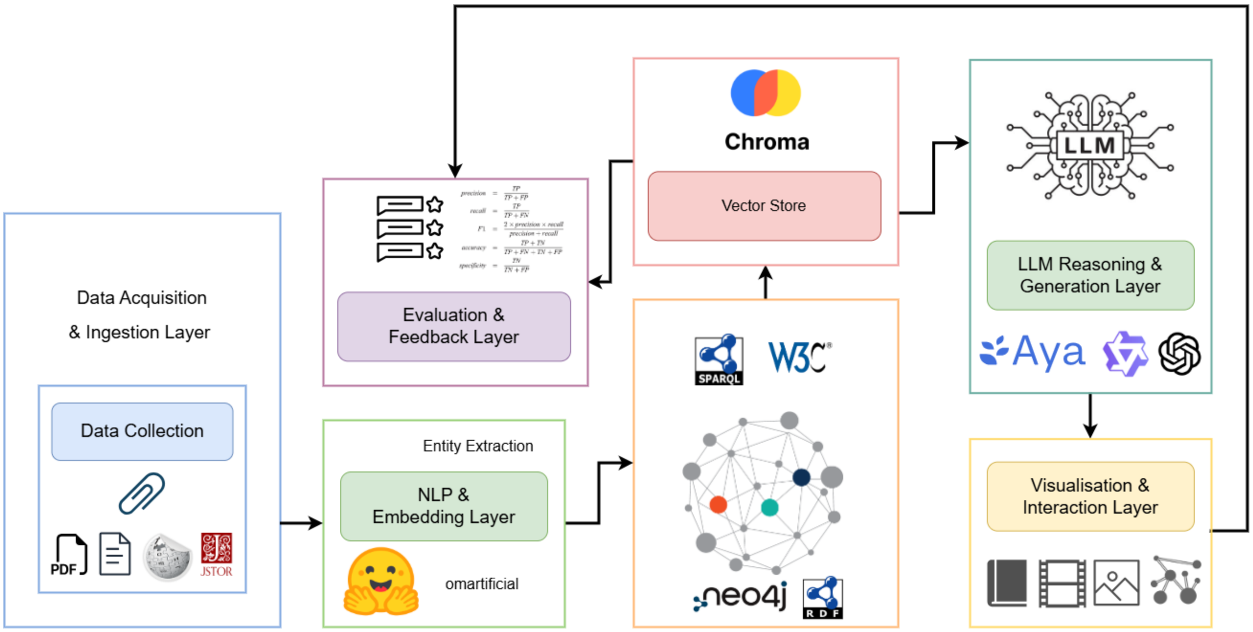


Figure 4: Initial_Knowledge_Graph_System_Design.png

This diagram represents the earliest conceptual design of the architecture. It serves as a baseline reference for how the system evolved.

Including this diagram provides design transparency and documents system maturity from concept to implementation-oriented architecture.

4.1.3 Sample Semantic KG System Architecture

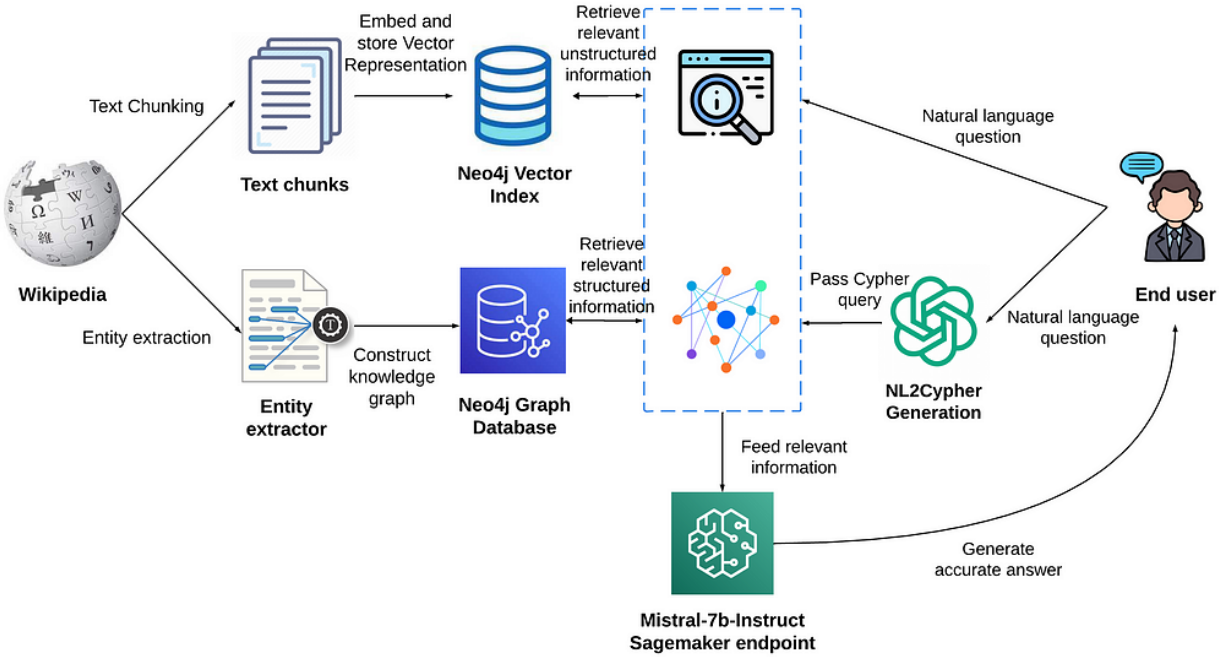

Text Chunking
Embed and store Vector Representation
Retrieve relevant unstructured information
Text chunks
Neo4j Vector Index
Natural language question
Wikipedia
Entity extraction
Retrieve relevant structured information
Pass Cypher query
End user
Natural language question
Construct knowledge graph
Entity extractor
Neo4j Graph Database
NL2Cypher Generation
Feed relevant information
Generate accurate answer
Mistral-7b-Instruct Sagemaker endpoint

Figure 5: Sample_Semantic_KG_System_Architecture.png

This sample illustrates the core functional architecture of knowledge graph generation, including acquisition, extraction, validation, and visualisation cycles, influenced from existing products.

This diagram is useful to communicate the solution to readers unfamiliar with knowledge graph processing.

4.1.4 Semantic Knowledge Graph Processing Pipeline

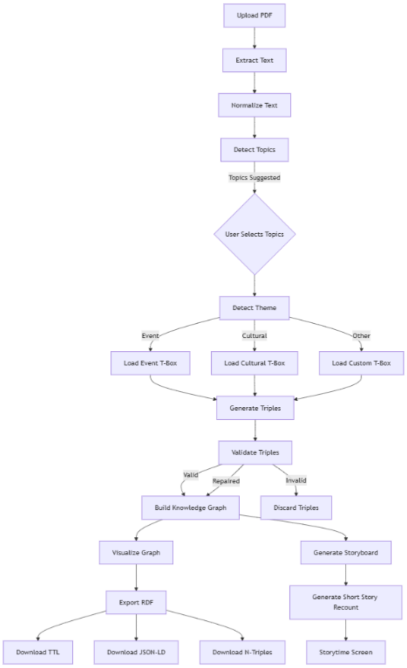


Figure 6: Semantic_KnowledgeGraph_Pipeline_Flowchart.md

This flowchart shows the end-to-end pipeline starting from PDF upload, topic selection, ontology constraint loading, triple generation, validation, and RDF export, including the *story tale* module. Story-time module is where the AI-generated storyboard and narrative recount derived from the KG will be located.

This diagram is needed to explain processing logic and data transformation flow.

4.2 Logical Model Design

Below shows a high-level and low-level design of the system:

4.2.1 UI Use-Case Flow

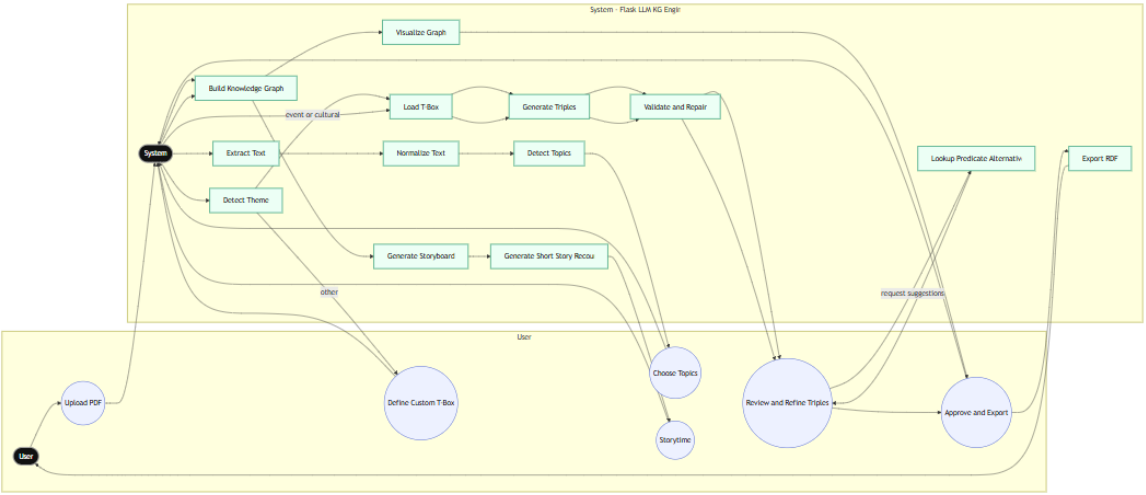


Figure 7: Use_Case_Diagram_(UI_Flowchart).md

This diagram represents the user interaction flow, showing how a human interacts with the system across stages such as:

- uploading documents
- selecting topics
- validating triples
- exporting RDF
- and viewing the knowledge graph

This is necessary because this explains human experience and interaction as a key design principle.

It also serves as the *Activity Diagram*, because it shows conditional routing, describes sequential system operations, includes human-in-the-loop review, and introduces Storytime narrative generation.

4.2.2 Physical Model Design

The physical model describes what the user expects to see and interact with. These are represented by the UI screen mock-ups.

Screenshots of the screens mock-up (UI_Mock-Up folder in GitHub) are found in Appendix D: UI Mock-up Design.

Chapter 5

Data Preprocessing

This chapter describes the data undergoing the ETL pipeline, including data collection, loading, profiling, prompting, and feature engineering. All scripts used for data manipulation and extraction are maintained and regularly updated in the associated GitHub repository:

https://github.com/Mohammad-ALADDASI/ChronoLearn

5.1 Data Collection and Description

Different structured, semi-structured, and unstructured textual data from multiple sources are used in the project to simulate the user-provided data content. The sample data includes books, scholarly research papers, and entries from online encyclopaedias, with a thematic focus on the histories and cultures of Jordan and Palestine. These documents vary in format, target audience, and narrative style, reflecting a range of potential user contexts.

Most documents do not include standardised metadata. Accordingly, manual annotation was carried out to assign labels for theme (e.g., culture-related or event-related), topic focus, and source type (e.g., book, scholarly paper, oral archive transcript, or webpage). This labelling supports downstream processing, classification, and KG construction.

5.1.1 "مدخل إلى الثقافة الوطنية والمدنية - Introduction to National Culture and Civilisation" Book by Abu Ruman et al.

This document is an educational book authored by Abu Ruman et al. published by the Ministry of Higher Education and Scientific Research and used in Jordanian universities as part of the compulsory National Education curriculum. The material of the data mostly contains event-based information as treaties, wars, declarations, accessions, etc. Therefore, this document input represents an event-themed pedagogical source of history used by teachers and facilitators for their students.

5.1.2 Jafra Song: Arabic Wikipedia Page

This document is an open-source produced online encyclopaedia entry from the Arabic-language Wikipedia platform. It presents descriptive, historical, and cultural information about the traditional song “Jafra,” including its origins, meanings, lyrical themes, and socio-cultural significance. Since it is a public, user-edited reference entry, the material combines narrative description with cultural interpretation rather than primary historical reporting. Therefore, this document input represents a secondary, community-generated culture-themed online informational source on Palestinian songs heritage.

5.1.3 "عودة إلى بقايا الأطلال: الذاكرة والذاكرة المولدة والتاريخ الحي في فلسطين / Return to Half-Ruins: Memory, Post memory, and Living History in Palestine"

This document is a scholarly work written in Arabic that examines themes of memory, post-memory, and lived or "living" history within the Palestinian context. The material is analytical and interpretive, focusing on personal, collective, and intergenerational experiences of displacement, place, and historical trauma. It therefore represents an academic, memory-themed cultural and event-related source that focuses on narrative, remembrance, and identity-formation.

5.2 Data, Loading, Profiling and Engineering

Documents and webpages are ingested using a combination of libraries and tools, including PyPDF2 and python-docx library for direct upload, BeautifulSoup and Selenium for scraping web-based content, particularly when access constraints were present for more restricted websites and saved as tokens when uploaded. Afterwards, text content is stored as tokenised input to support following processing steps.

Initially, the data is pre-processed manually by normalising the Arabic letters such as removing Tatweel, normalising Alef, Yaa' and Ta Marbutah variants, and removing diacritics (harakat). These minimal preprocessing steps prepared the data for NER directly using LLM API calls and enhanced the prompts' outputs.

5.3 Feature Engineering

For construction of the KG, LLMs were employed exclusively for NLP processes and NER to support triple extraction. This design choice enables the system to scale to large and heterogeneous input corpora, while accommodating unstructured and unpredictable language through flexible prompt engineering unlike manual NLP methods. The pipeline also includes thematic extraction, where texts are categorised according to either event-related or cultural themes; if a theme cannot be automatically inferred, it is instead specified through a human-in-the-loop process. Additionally, topics of interest that guide the extraction process within a default T-Box schema associated with each theme, are extracted via LLM and selected by the user too with human-in-the-loop process. This schema supports systematic KG construction and facilitates subsequent named NED following NER.

Appendix A

Users' Manual

To be determined after the implementation of the proposal.

Appendix B

Document Changes

These changes are to be discussed after the implementation of the proposal in phase 2 of the project.

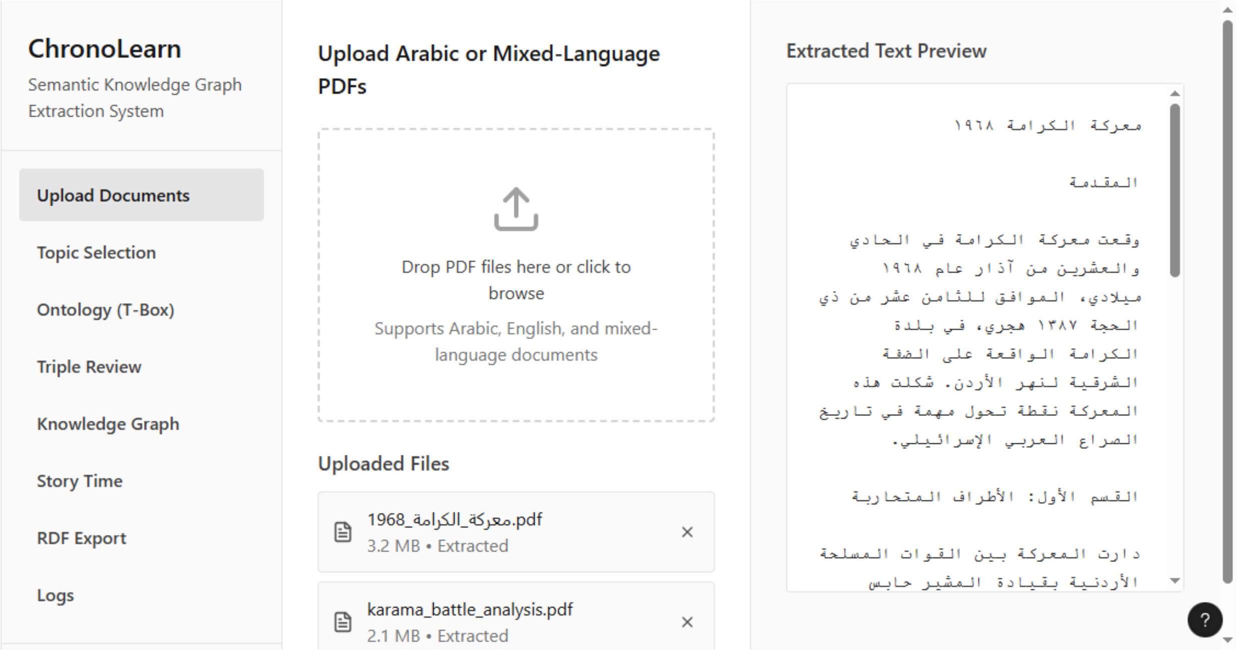
Appendix C

Code Documentation

These changes are to be discussed after the implementation of the proposal in phase 2 of the project.

Appendix D

UI Mock-up Design



Upload Arabic or Mixed-Language PDFs

Drop PDF files here or click to browse

Supports Arabic, English, and mixed-language documents

Uploaded Files

1968_معركة_الكرامة.pdf
3.2 MB • Extracted

karama_battle_analysis.pdf
2.1 MB • Extracted

Extracted Text Preview

معركة الكرامة ١٩٦٨

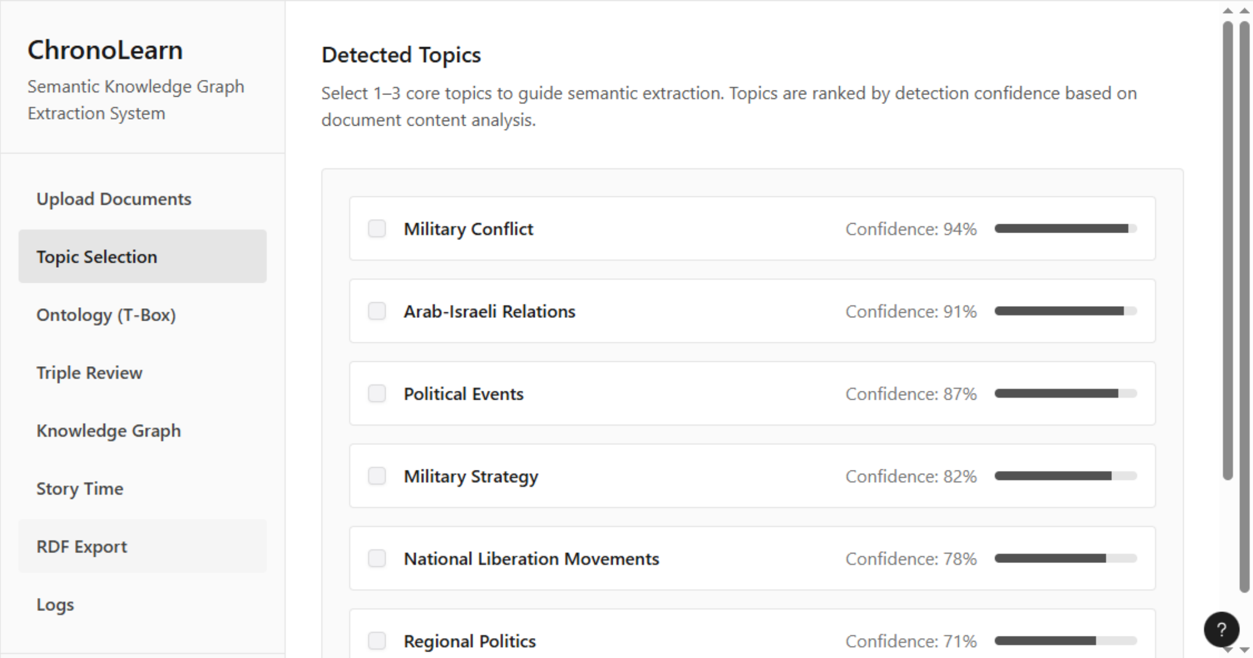
المقدمة

وقعت معركة الكرامة في الحادي
والعشرين من آذار عام ١٩٦٨
ميلادي، الموافق للثامن عشر من ذي
الحجة ١٣٨٧ هجري، في بلدة
الكرامة الواقعة على الضفة
الشرقية لنهر الأردن. شكلت هذه
المعركة نقطة تحول مهمة في تاريخ
الصراع العربي الإسرائيلي.

القسم الأول: الأطراف المتحاربة

دارت المعركة بين القوات المسلحة
الأردنية بقيادة المشير حابس

Figure 8: 1_Upload_Documents_Screen.png

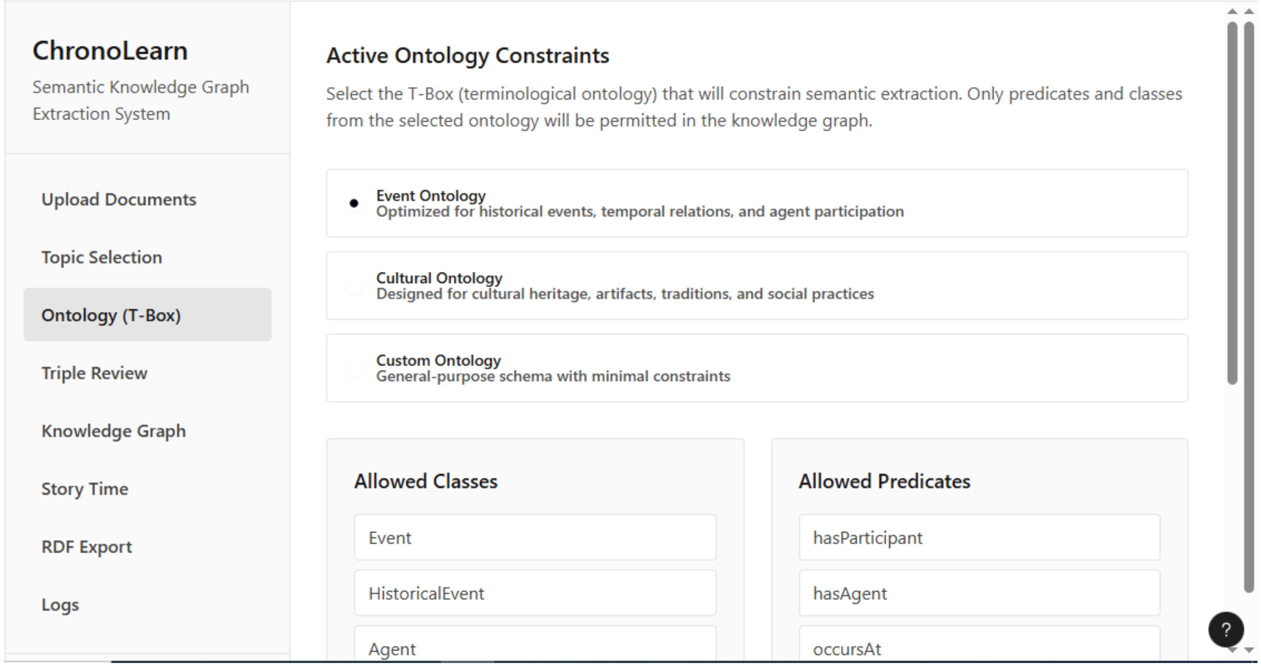


Detected Topics

Select 1–3 core topics to guide semantic extraction. Topics are ranked by detection confidence based on document content analysis.

- □ Military Conflict — Confidence: 94%
- □ Arab-Israeli Relations — Confidence: 91%
- □ Political Events — Confidence: 87%
- □ Military Strategy — Confidence: 82%
- □ National Liberation Movements — Confidence: 78%
- □ Regional Politics — Confidence: 71%

Figure 9: 2_Topic_Selection.png



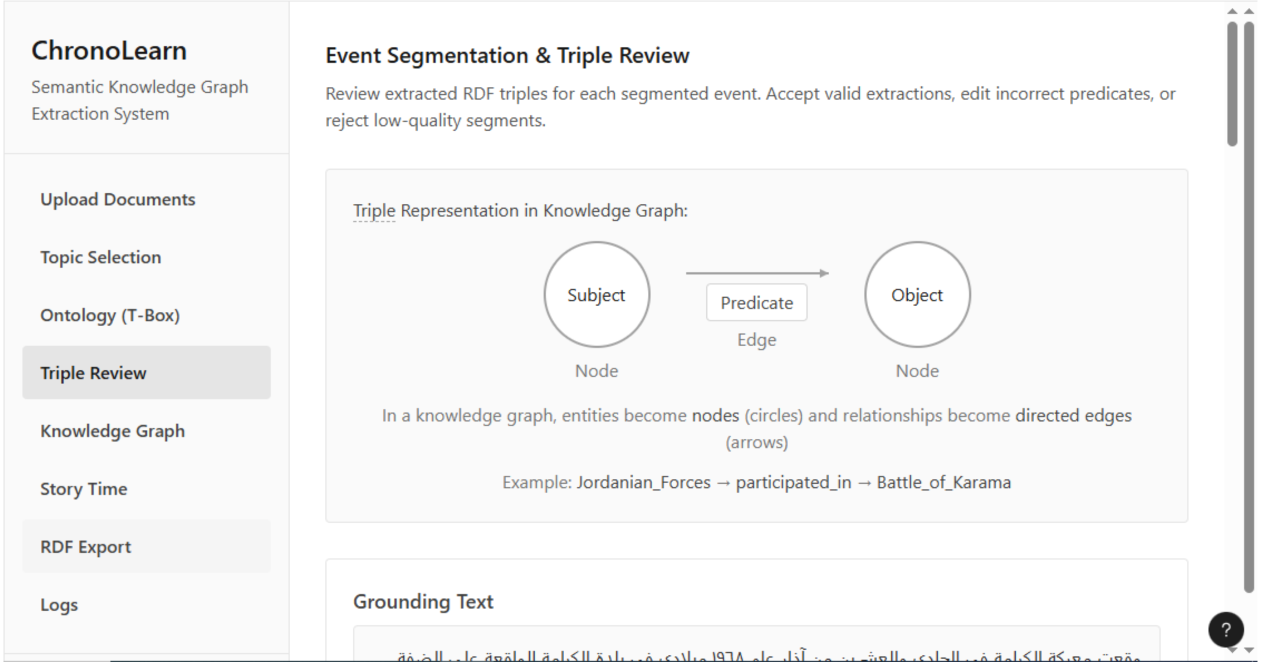
Active Ontology Constraints

Select the T-Box (terminological ontology) that will constrain semantic extraction. Only predicates and classes from the selected ontology will be permitted in the knowledge graph.

- ● **Event Ontology** Optimized for historical events, temporal relations, and agent participation
- ○ **Cultural Ontology** Designed for cultural heritage, artifacts, traditions, and social practices
- ○ **Custom Ontology** General-purpose schema with minimal constraints

Allowed Classes	Allowed Predicates
Event	hasParticipant
HistoricalEvent	hasAgent
Agent	occursAt

Figure 10: 3_Ontology-TBOX.png



Event Segmentation & Triple Review

Review extracted RDF triples for each segmented event. Accept valid extractions, edit incorrect predicates, or reject low-quality segments.

Triple Representation in Knowledge Graph:

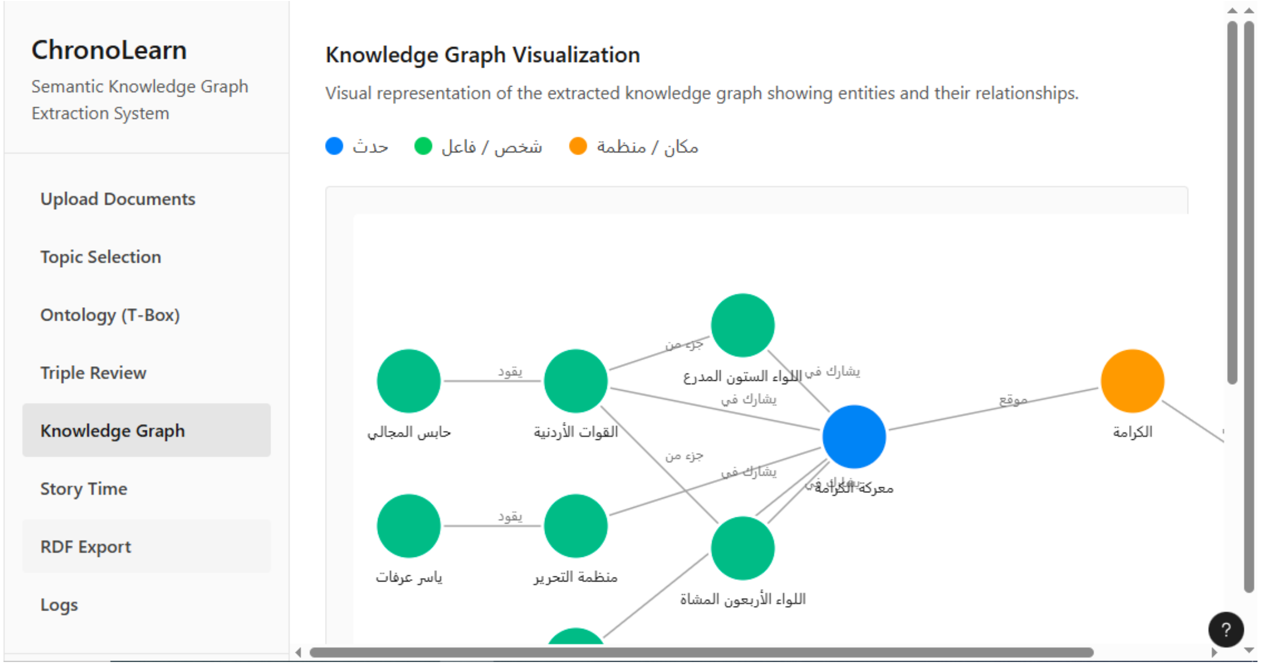
In a knowledge graph, entities become nodes (circles) and relationships become directed edges (arrows)

Example: Jordanian_Forces → participated_in → Battle_of_Karama

Grounding Text

وقعت معركة الكرامة في الحادي والعشرين من آذار عام ١٩٦٨ ميلادي في بلدة الكرامة الواقعة على الضفة

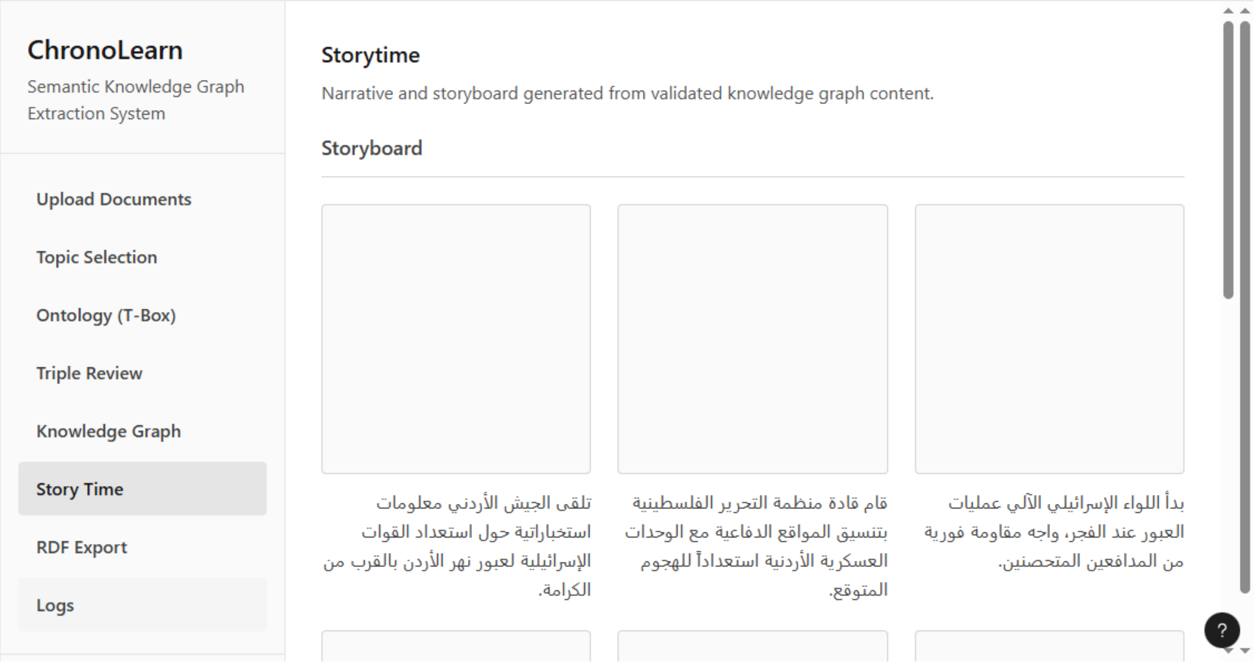
Figure 11: 4_Triple_Review.png



Knowledge Graph Visualization

Visual representation of the extracted knowledge graph showing entities and their relationships.

Figure 12: 5_Knowledge_Graph_Visualisation.png



Storytime

Narrative and storyboard generated from validated knowledge graph content.

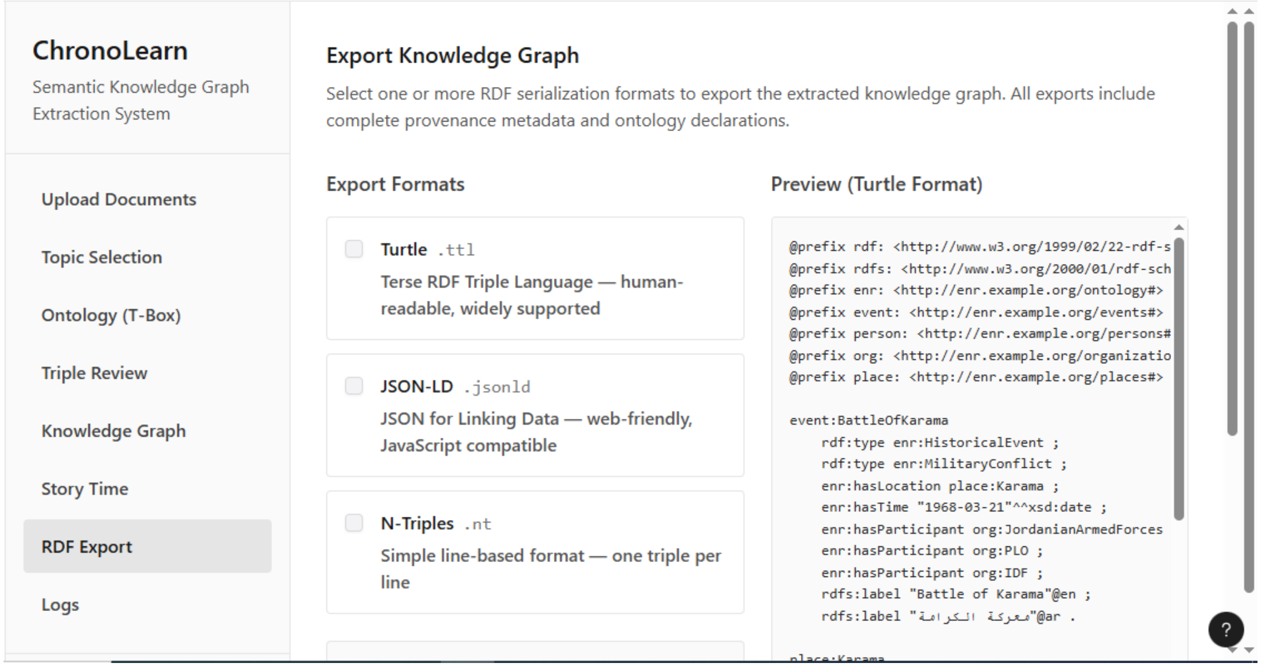
Storyboard

تلقى الجيش الأردني معلومات استخباراتية حول استعداد القوات الإسرائيلية لعبور نهر الأردن بالقرب من الكرامة.

قام قادة منظمة التحرير الفلسطينية بتنسيق المواقع الدفاعية مع الوحدات العسكرية الأردنية استعداداً للهجوم المتوقع.

بدأ اللواء الإسرائيلي الآلي عمليات العبور عند الفجر، واجه مقاومة فورية من المدافعين المتحصنين.

Figure 13: 6_Story_Time.png



Export Knowledge Graph

Select one or more RDF serialization formats to export the extracted knowledge graph. All exports include complete provenance metadata and ontology declarations.

Export Formats

☐ Turtle .ttl
Terse RDF Triple Language — human-readable, widely supported

☐ JSON-LD .jsonld
JSON for Linking Data — web-friendly, JavaScript compatible

☐ N-Triples .nt
Simple line-based format — one triple per line

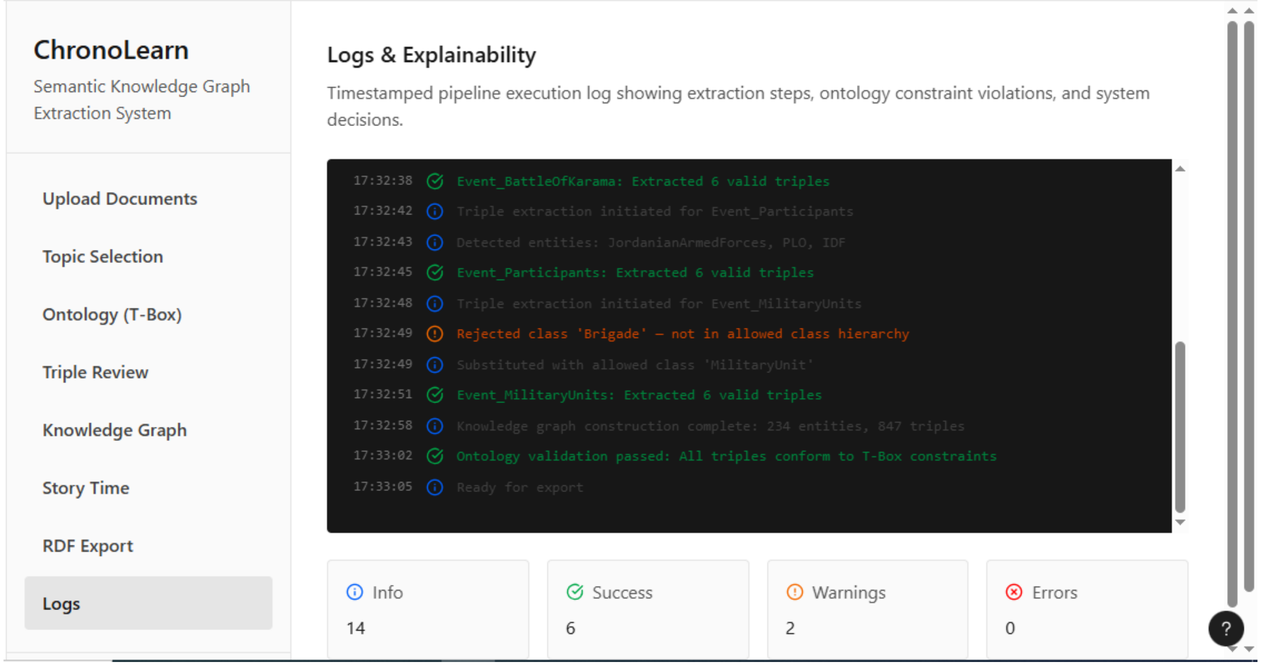
Preview (Turtle Format)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-s
@prefix rdfs: <http://www.w3.org/2000/01/rdf-sch
@prefix enr: <http://enr.example.org/ontology#>
@prefix event: <http://enr.example.org/events#>
@prefix person: <http://enr.example.org/persons#
@prefix org: <http://enr.example.org/organizatio
@prefix place: <http://enr.example.org/places#>

event:BattleOfKarama
    rdf:type enr:HistoricalEvent ;
    rdf:type enr:MilitaryConflict ;
    enr:hasLocation place:Karama ;
    enr:hasTime "1968-03-21"^^xsd:date ;
    enr:hasParticipant org:JordanianArmedForces
    enr:hasParticipant org:PLO ;
    enr:hasParticipant org:IDF ;
    rdfs:label "Battle of Karama"@en ;
    rdfs:label "معركة الكرامة"@ar .

place:Karama
```

Figure 14: 7_RDF_Export.png



Logs & Explainability

Timestamped pipeline execution log showing extraction steps, ontology constraint violations, and system decisions.

```
17:32:38 ✓ Event_BattleOfKarama: Extracted 6 valid triples
17:32:42 ⓘ Triple extraction initiated for Event_Participants
17:32:43 ⓘ Detected entities: JordanianArmedForces, PLO, IDF
17:32:45 ✓ Event_Participants: Extracted 6 valid triples
17:32:48 ⓘ Triple extraction initiated for Event_MilitaryUnits
17:32:49 ! Rejected class 'Brigade' — not in allowed class hierarchy
17:32:49 ⓘ Substituted with allowed class 'MilitaryUnit'
17:32:51 ✓ Event_MilitaryUnits: Extracted 6 valid triples
17:32:58 ⓘ Knowledge graph construction complete: 234 entities, 847 triples
17:33:02 ✓ Ontology validation passed: All triples conform to T-Box constraints
17:33:05 ⓘ Ready for export
```

ⓘ Info	✓ Success	! Warnings	⊗ Errors
14	6	2	0

Figure 15: 8_Logs.png

References

[1] E. A. Meguid and M. Collins, “Students’ perceptions of lecturing approaches: Traditional versus interactive teaching,” *Advances in Medical Education and Practice,* vol. 8, p. 229–241, 2017.

[2] Z. Zhang and H. Soh, “Extract, Define, Canonicalize: Towards Efficient Large-Scale Knowledge Graph Construction with LLMs,” 2024.

[3] P. Mohanty, “EduEmbedd: An Educational Knowledge Graph Embedding Framework for Pedagogical Content Linking,” in *CEUR Workshop Proceedings*, 2023.

[4] M. Wang, Y. Gao, M. Fang and X. Lyu, “ChatWeaver: An Interactive LLM-Supported Knowledge Graph System,” in *IEEE International Conference on Artificial Intelligence in Information and Networks (AINIT)*, 2025.

[5] Q. Zhang, J. Dong, H. Chen, D. Zha, Z. Yu and X. Huang, “KnowGPT: Knowledge Graph based Prompting for Large Language Models,” *Advances in Neural Information Processing Systems,* vol. 37, p. 6052–6080, 12 2024.

[6] Ö. F. Akgül, F. Zhu, Y. Yang, R. Kannan and V. Prasanna, “RECIPE-TKG: From Sparse History to Structured Reasoning for LLM-based Temporal Knowledge Graph Completion,” 2025.

[7] Y. Yan, Y. Hou, Y. Xiao, R. Zhang and Q. Wang, “KNowNEt: Guided Health Information Seeking from LLMs via Knowledge Graph Integration,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 31, no. 1, p. 547–557, 1 2025.

[8] D. Muralidharan, “Knowledge Graph-based Multiple-Choice Question Generation,” January 2024.

[9] J. Yang, “Integrated Application of LLM Model and Knowledge Graph in Medical Text Mining and Knowledge Extraction,” *Social Medicine and Health Management,* 2024.

[10] Y. Sun, W. Yang and Y. Liu, “The Application of Constructing Knowledge Graph of Oral Historical Archives Resources Based on LLM-RAG,” in *Proceedings of the 2024 8th International Conference on Information System and Data Mining (ICISDM ’24)*, New York, NY, USA, 2024.