มหาวิทยาลัยอัสสัมชัญ
ASSUMPTION UNIVERSITY
of THAILAND

VINCENT MARY
SCHOOL of ENGINEERING, SCIENCE and TECHNOLOGY

Senior Project Report

AUREX - Exam Paper Generation System

Zwe La Pyae,
Khant Min Lwin,
Thet Myat Noe Thwin

Asst. Prof. Dr. Rachsuda Setthawong (Advisor)

CSX 3010 Senior Project 1 (1/2025)

ITX 3010 Senior Project 1 (1/2025)

Senior Project Approval

Project title:	AUREX - Exam Paper Generation System
Academic Year:	1/2025
Authors:	Zwe La Pyae (6530207) Khant Min Lwin (6530203) Thet Myat Noe Thwin (6530213)
Project Advisor:	Asst. Prof. Dr. Rachsuda Setthawong

The Senior Project committee's cooperation between the Department of Computer Science and Information Technology, Vincent Mary School of Engineering, Science and Technology, Assumption University had approved this Senior Project. The Senior Project in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science and Information technology.

Approval Committee:

…………………………..
(Asst. Prof. Dr. Rachsuda Setthawong)
Project Advisor

…………………………..
(Asst. Prof. Dr. Paitoon Porntrakoon)
Committee Member

…………………………..
(Dr. Phyo Min Tun)
Committee Member

Abstract

Assessment plays a critical role in higher education, but exam preparation is often fragmented and inefficient. In large courses with multiple sections, each lecturer submits questions to a course coordinator, who must consolidate them into a single fair and balanced exam. This process is repetitive, error-prone, and time-consuming, particularly when handling large question pools across multiple subjects.

AUREX is designed to address these challenges by providing an intelligent exam paper generation system. It enables coordinators to integrate Excel-based question banks from multiple lecturers, detect duplicates and overlaps, balance coverage across topics, and format the final exam consistently using Jinja2-based Word templates. With features such as grammar checks, similarity detection, pin-and-shuffle editing, and Word export, AUREX reduces coordinator workload while preserving flexibility and academic control. The system runs locally to ensure privacy and is tailored to the real exam management needs of universities.

Table Of Contents

Table Of Figures

Table Of Tables

Chapter 1: Introduction

1.1 Background & Motivation

At Assumption University, many general education (GE) and business (BBA) courses are taught in multiple sections by different lecturers. Each lecturer is responsible for preparing exam questions, which must then be submitted to the course coordinator. The coordinator's role is to merge these inputs into a final exam paper that is fair, balanced, and non-repetitive across chapters and categories.

This coordination process is challenging: the coordinator must review numerous contributions, manually check for duplicates, and ensure consistent formatting across the entire paper. As course enrollments grow and the number of lecturers increases, the complexity of exam assembly becomes overwhelming.

AUREX was developed to specifically reduce the coordinator's workload by automating repetitive tasks such as duplication detection, formatting, and randomization, while still allowing human oversight and control.

1.2 Problem Statement

In many academic programs, particularly in business and general education courses, examinations are prepared collaboratively by multiple lecturers. Each lecturer submits questions from their assigned chapters or topics, which are then compiled and finalized by the **course coordinator**. This process is often **time-consuming, repetitive, and error-prone**, as coordinators must manually:

- Consolidate large volumes of questions from different contributors.
- Ensure balanced coverage of topics, chapters, and question types.
- Avoid duplication or overly similar questions across exams.
- Format exam papers consistently according to institutional requirements.

The lack of automation increases the workload for course coordinators, especially in large classes with many sections. This not only slows down the exam preparation process but also increases the risk of inconsistencies and human error.

1.3 Objectives

The objectives of this project are to:

1. **Automate Exam Paper Generation** – Reduce the manual effort required to compile exam papers by automatically selecting and formatting questions.
2. **Ensure Balanced Question Distribution** – Provide mechanisms to distribute questions fairly across chapters, categories, and types.
3. **Support Customizable Templates** – Enable flexible formatting of exam papers through Word templates with Jinja2 integration.
4. **Enhance Usability for Course Coordinators** – Offer editing tools such as pinning, shuffling, and similarity detection to improve exam quality.
5. **Maintain Data Privacy and Institutional Compliance** – Operate entirely on local machines without external connections, minimizing data security risks.

1.4 Scope & Limitations of the Project

Scope:

The system is designed for **course coordinators** responsible for compiling exams from multiple lecturers' submitted questions. Key features include:

1. Uploading question banks in Excel format.
2. Uploading exam templates in Word format using Jinja2 placeholders.
3. Automated question selection and balanced distribution across chapters and types.
4. Editing tools such as pinning questions, shuffling order, and duplication detection.
5. Previewing and exporting finalized exam papers in Word format.
6. Local-only processing to protect academic confidentiality.

Limitations:

While effective, the current implementation has certain constraints:

1. **Single User Role** – Supports only course coordinators as primary users.
2. **Limited Question Types** – Supports five core question types (MCQ, true/false, short answer, matching, essay).
3. **No Persistent Database** – To prioritize privacy, the system does not store uploaded data centrally.
4. **Offline-First Design** – Runs locally without cloud integration, meaning no collaborative editing or online storage is available.

Chapter 2: Related Work

2.1 Manual and Semi-Automated Exam Preparation

Traditionally, exam paper preparation has relied heavily on manual methods, with lecturers compiling questions in **Word documents** or **Excel spreadsheets**. Coordinators often merge multiple files, copy-paste questions, adjust numbering, and apply formatting manually. This approach is **time-consuming, repetitive, and error-prone**, especially in courses with many sections and lecturers contributing questions.

Semi-automated solutions, such as spreadsheets with basic macros or partially formatted Word templates, reduce some manual work but still require **significant human oversight**. These methods also lack built-in duplication detection, balanced question distribution, or scalability for large institutions.

2.2 Existing Tools and LMS Approaches

Several digital tools and learning management systems (LMS) have attempted to address exam preparation challenges:

- **Learning Management Systems (e.g., Moodle, Canvas, Blackboard):**
  LMS platforms provide quiz modules and test banks for online delivery. While effective for digital assessments, they are **not optimized for printed exam papers** that require strict formatting, institutional templates, and offline availability.

- **Online Question Generators:**
  Tools such as simple MCQ generators or AI-based quiz creators can automatically generate objective-type questions. However, these are often **limited in scope**, focusing mainly on multiple-choice formats with minimal flexibility for editing or adapting to institutional standards.

- **Institution-Specific Software:**
  Some universities develop in-house exam management systems, but these are often tightly coupled to their LMS or require online access, raising concerns about **privacy and data security** when handling sensitive examination material.

2.3 Gaps and Opportunities

Despite the availability of these tools, several gaps remain unaddressed in the context of **academic paper-based exam preparation**:

1. **Flexible Word Template Rendering** – Few systems allow course coordinators to fully customize exam layout and formatting through reusable Word templates.
2. **Local Privacy and Security** – Most online platforms require data upload to external servers, which may not comply with institutional privacy policies.
3. **Duplicate and Similarity Detection** – Existing systems lack **NLP-driven checks** to detect similar or repeated questions across multiple exam sessions.
4. **Coordinator-Oriented Editing Tools** – Features like **pinning key questions, shuffling order, and balancing distribution** are rarely integrated into one cohesive tool.

AUREX directly addresses these gaps by combining automation, NLP-based duplication detection, template-driven Word export, and strict offline operation. This unique combination makes it particularly suitable for institutions that require **scalable, private, and customizable exam generation workflows**.

Chapter 3: Proposed Methodology

3.1 System Workflow Overview

The proposed system, **AUREX**, is designed to streamline the exam paper generation process through a structured, semi-automated workflow. The key stages are as follows:

1. **Upload Phase**
   - The course coordinator uploads an **Excel-based question bank** containing multiple question types.
   - A **Word template** containing Jinja2 placeholders is also uploaded or selected from predefined templates.
2. **Quality Assurance**
   - The course coordinator may choose whether to apply grammar and duplication checks during processing.
3. **Parsing and Categorization**
   - The backend parses the Excel file and converts each entry into structured JSON objects.
   - Metadata such as question type, chapter, answer choices, and solutions are extracted for later filtering and organization.
4. **Question Selection**
   - Based on the input, the system retrieves random sets of questions per category.
5. **Editing Stage**
   - An integrated editor allows coordinators to refine the retrieved set.
   - Features include **pinning/unpinning** questions, **shuffling** non-pinned items, and resolving duplication flags.
6. **Preview and Export**
   - Coordinators preview the exam paper.
   - The finalized exam paper is rendered into a **fully formatted Word document** based on the template.

3.2 Data Processing Pipeline

The backend implements a lightweight **data processing pipeline**:

- **Excel Parsing**: Each row of the uploaded question bank is parsed into a JSON object representing question text, metadata, and optional media.
- **Normalization**: Preprocessing includes lowercasing, punctuation handling, and tokenization to support NLP-based operations.
- **Temporary Storage**: Session-specific local folders store uploaded files. A persistent database is intentionally excluded to maintain exam data privacy.

3.3 Grammar Error Detection

The **Grammar Checking System** constitutes a critical component within the AUREX architecture, designed to ensure linguistic precision and overall quality assurance of examination materials. The module automatically identifies and tags questions with grammatical inconsistencies.

3.3.1 System Architecture

The module integrates the **LanguageTool framework** through a local instance, enabling comprehensive grammar and spelling verification without reliance on external APIs. The implementation utilizes the **language_tool_python** library for seamless communication with the LanguageTool server. This configuration supports **offline, privacy-preserving operation**, ensuring that examination content remains fully confidential throughout the analysis process.

The system operates under a **multi-layered processing model**. While the framework supports multiple languages, its primary configuration focuses on English grammar and syntax, reflecting the linguistic context of the examination materials.

3.3.2 Error Detection Framework

The grammar checking mechanism is structured around four principal categories of linguistic evaluation:

- **Spelling Errors:** Identification of typographical mistakes, misused word forms, and unrecognized lexical items.

- **Grammatical Violations:** Detection of syntactic inconsistencies such as subject–verb disagreement, tense mismatches, and incomplete sentence structures.
- **Punctuation and Capitalization Issues:** Recognition of missing or incorrect punctuation, inconsistent capitalization, and improper sentence delimiters.
- **Stylistic Inconsistencies:** Flagging of redundant phrases, ambiguous constructions, and deviations from formal academic writing norms.

Each identified error is classified according to its severity and type, enabling structured presentation and prioritized correction.

3.4 Question Duplication and Similarity Detection

The Duplicate and Similarity Detection Module in AUREX is designed to identify and eliminate semantically similar or duplicate questions within large academic question banks. The system employs a **multi-tiered hybrid approach** combining lexical, semantic, and structural similarity measures to ensure both efficiency and precision.

3.4.1 Multi-Tired Detection Architecture

The system operates through three sequential tiers:

- **Tier 1 – Lexical Pre-Filtering (TF-IDF Similarity):**

  In the initial stage, all questions are vectorized using the Term Frequency–Inverse Document Frequency (TF-IDF) model with unigram and bigram features. Cosine similarity is then calculated between all question pairs, and pairs with similarity above a threshold (0.3) are selected as potential candidates for further analysis. This step rapidly reduces the computational space, retaining only those pairs likely to be duplicates.

- **Tier 2 – Detailed Multi-Modal Similarity Analysis:**

  The shortlisted pairs undergo a comprehensive analysis that integrates multiple linguistic perspectives:
  - **Exact Similarity:** Jaccard coefficient computed on normalized word sets to capture lexical overlap.
  - **Semantic Similarity:** Sentence embeddings generated via the SentenceTransformer model (all-MiniLM-L6-v2) capture conceptual equivalence beyond surface wording.

    - ○ **TF-IDF Similarity:** Fine-grained cosine similarity of re-vectorized text pairs ensures contextual alignment.
    - ○ **Keyword Similarity:** Comparison of stemmed and filtered keywords captures thematic consistency.

  These similarity components are fused into a single score through a weighted aggregation model (40% semantic, 30% exact, 20% TF-IDF, 10% keyword), ensuring balanced sensitivity to both surface and semantic resemblance.

- **Tier 3 – Clustering and Representative Selection:**

  Using a Union-Find clustering mechanism, all questions exceeding the final similarity threshold (0.6) are grouped into duplicate clusters. Within each cluster, the system automatically designates the most complete and informative question as the representative. Completeness is determined based on question length, presence of answer options, detailed explanations, and availability of metadata or media.

3.4.2 Text Normalization and Keyword Extraction

All text content undergoes a rigorous normalization process to standardize linguistic structure before analysis. The preprocessing pipeline includes lowercasing, whitespace normalization, punctuation removal (except question marks), tokenization, stop word elimination, and stemming using the Porter Stemmer algorithm. Additionally, keywords are extracted from both the question and its associated answer options to ensure comprehensive content representation.

3.5 Proposed Interactive Editor Features

The proposed AUREX Interactive Editor is designed to function as the central workspace through which course coordinators can curate and refine automatically generated question sets into finalized, examination-ready documents. The editor will serve as a bridge between automated question generation and human academic judgment, supporting coordinators with intelligent tools that enhance assessment quality, maintain fairness, and streamline the compilation process.

3.5.1 Dynamic Question Management

Pin/Unpin System

The proposed editor will incorporate a **pin/unpin mechanism** that enables coordinators to designate essential questions as fixed within the exam structure. Pinned questions will remain static throughout reshuffling and regeneration processes, ensuring that key assessment objectives and mandatory curriculum content are preserved. This approach aims to maintain the pedagogical integrity of the examination while still allowing flexibility in question selection.

This functionality will be particularly beneficial for:

- Questions addressing core learning outcomes or accreditation requirements
- Validated questions with proven reliability and difficulty balance
- Sequential or dependent questions requiring fixed order
- Coordinator-defined “must-include” items from previous exams

Shuffle Mechanism

To complement the pinning system, a **shuffle mechanism** will be implemented. Unlike traditional randomization, this system will utilize metadata attributes—such as question type, chapter mapping, and difficulty level—to replace or reorder questions intelligently. The proposed algorithm will ensure that when replacements are made, they maintain the exam’s intended structure, ensuring balanced topic coverage and consistency in assessment depth.

3.5.2 Duplication and Grammar Detection Interface

Duplication Indicators

An integrated **visual similarity analysis system** that translates complex duplicate detection outputs into a user-friendly interface. Potential duplicates will be visually highlighted using a tiered color-coding system, guiding coordinators’ attention to areas that require human review.

The interface will categorize similarities into three levels:

- **High-risk duplicates**: flagged prominently with suggested replacements
- **Moderate similarities**: displayed with contextual metadata for manual review
- **Related clusters**: visually grouped to help coordinators identify thematic overlaps

This design intends to enhance situational awareness while minimizing cognitive load, allowing coordinators to make quick, informed decisions without requiring technical expertise in similarity metrics.

Potential Grammar Error Indicators

The proposed system will integrate **grammar and linguistic validation tools** to support content quality assurance. The methodology emphasizes non-intrusive design, where potential language issues are displayed through subtle indicators. Coordinators may expand these indicators to fix these issues, thereby maintaining workflow continuity.

3.6 ROI Analysis

Although AUREX was developed without direct financial cost—utilizing open-source tools and in-house development—its return on investment is demonstrated through **efficiency, scalability, and academic value** rather than monetary gain.

Time Efficiency

- AUREX significantly reduces exam paper preparation time.
- Traditional manual compilation can take several hours or even days per course; AUREX automates question selection, formatting, and duplication checks, cutting this process to minutes.

Error Reduction

- The integrated grammar and duplication checks minimize human oversight.
- Teachers and coordinators can detect repeated or inconsistent questions before publishing, improving exam quality and fairness.

Operational Efficiency

- The system centralizes question management for multiple lecturers contributing to the same course.
- Course coordinators can easily randomize questions from different sections without redundant effort.

Academic and Long-Term Value

- By reducing repetitive administrative work, lecturers can focus more on pedagogy and content quality.
- The project demonstrates a scalable framework for future digital examination workflows within the university.

Chapter 4: Design of The System

4.1 Functional Requirements Specification

4.1.1 Stakeholders

We classified the stakeholders into 2 categories. Direct Stakeholders are whom our system will be designed for and has a direct use. Indirect users are the parties that are affected by the system.

- Direct Stakeholders - Course Coordinators
- Indirect Stakeholders - Teachers, Developers, Students

4.1.2 Actors and Goals

Primary Actor - *Course Coordinator*

The Course Coordinator is the sole direct user of the Exam Paper Generation System. This individual is responsible for compiling, organizing, and finalizing examination papers using question banks submitted by lecturers in Excel format.

Goals:

1. **Automate Exam Compilation:** Streamline the process of assembling exam papers from multiple Excel files, minimizing manual effort and reducing the risk of human error.
2. **Ensure Balanced Question Distribution:** Guarantee fair and comprehensive coverage of chapters and question types in the generated examination papers.
3. **Maintain Consistent Formatting:** Apply institutional formatting standards to exam papers through the use of customizable Word templates.
4. **Edit and Refine Exam Content:** Provide tools for pinning, shuffling, and managing questions to enhance the quality and integrity of the exam.
5. **Detect Duplicates and Errors:** Identify and highlight duplicate or similar questions, as well as grammar issues, during the processing stage.
6. **Protect Data Privacy:** Ensure all data processing occurs locally, maintaining academic confidentiality and compliance with institutional policies.

4.1.3 Use Cases

1. **Upload Question Bank:**

   The Course Coordinator uploads an Excel file containing the question bank for processing.

2. **Download Excel Template:**

   The Course Coordinator downloads a standardized Excel template to ensure proper formatting of question submissions.

3. **Download Word Template:**

   The Course Coordinator downloads a standardized Word template containing pre-defined Jinja2 placeholders to ensure consistent structure and formatting for exam papers.

4. **Upload Word Template:**

   The Course Coordinator uploads a Word template to customize the formatting of the exam paper.

5. **Select and Manage Word Templates:**

   The Course Coordinator selects, sets as default, or deletes Word templates for exam generation.

6. **Configure Processing Options:**

   The Course Coordinator enables or disables duplicate detection and grammar checking during question processing.

7. **Edit Question Selection:**

   The Course Coordinator pins or shuffles questions to adjust the exam content prior to finalization.

8. **Preview and Export Exam Paper:**

   The Course Coordinator previews the compiled exam and exports the finalized version in Word format.

9. **Check Old Question Similarity:**

The Course Coordinator checks the generated question and compares it with different questions from the past/previous semesters to make each question paper unique.

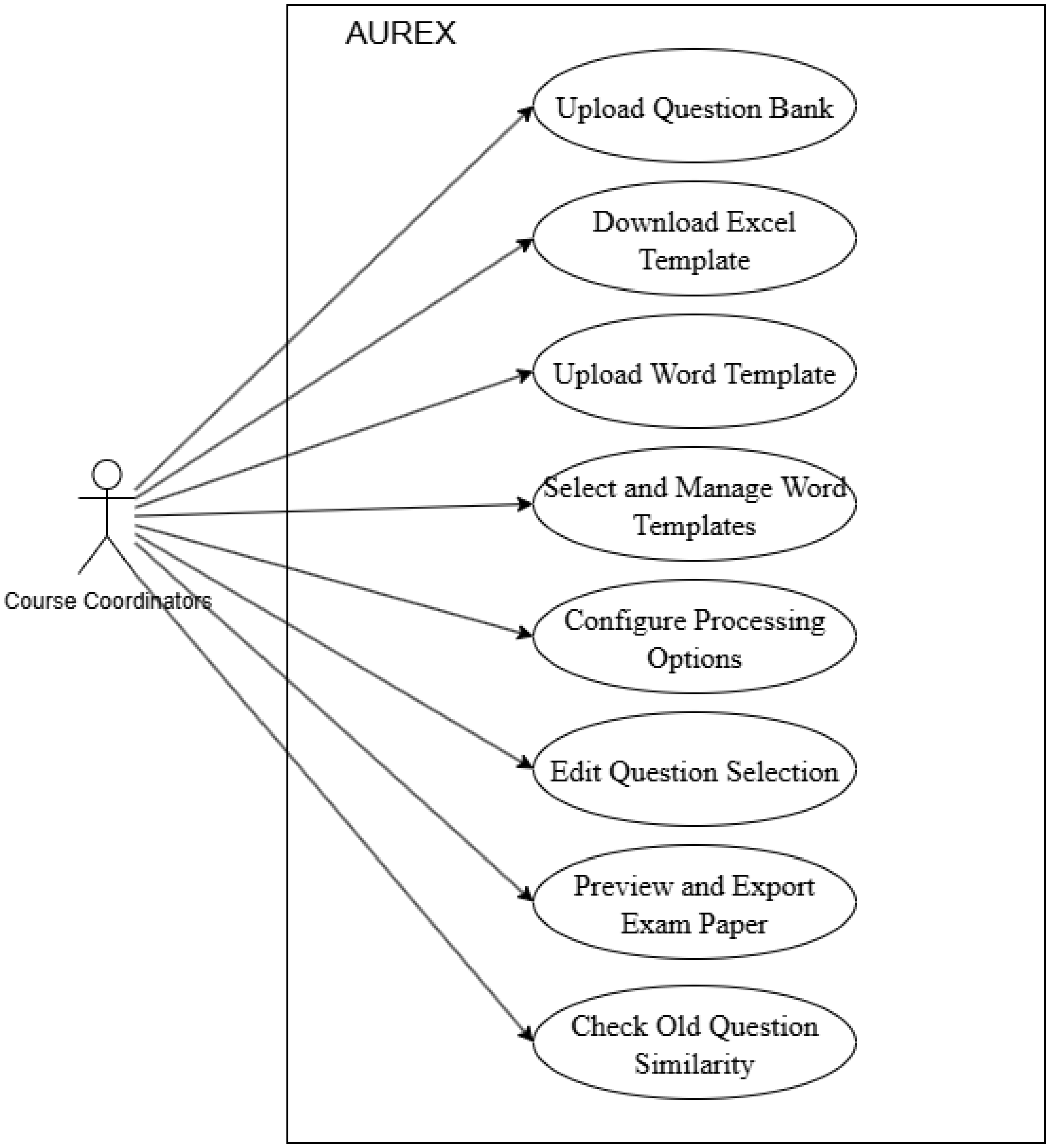

AUREX
Upload Question Bank
Download Excel Template
Upload Word Template
Select and Manage Word Templates
Configure Processing Options
Edit Question Selection
Preview and Export Exam Paper
Check Old Question Similarity
Course Coordinators

Figure 4. 1: Use Case Diagram of the AUREX - Exam Paper Generation System

4.2 System Design

4.2.1 Process Diagram

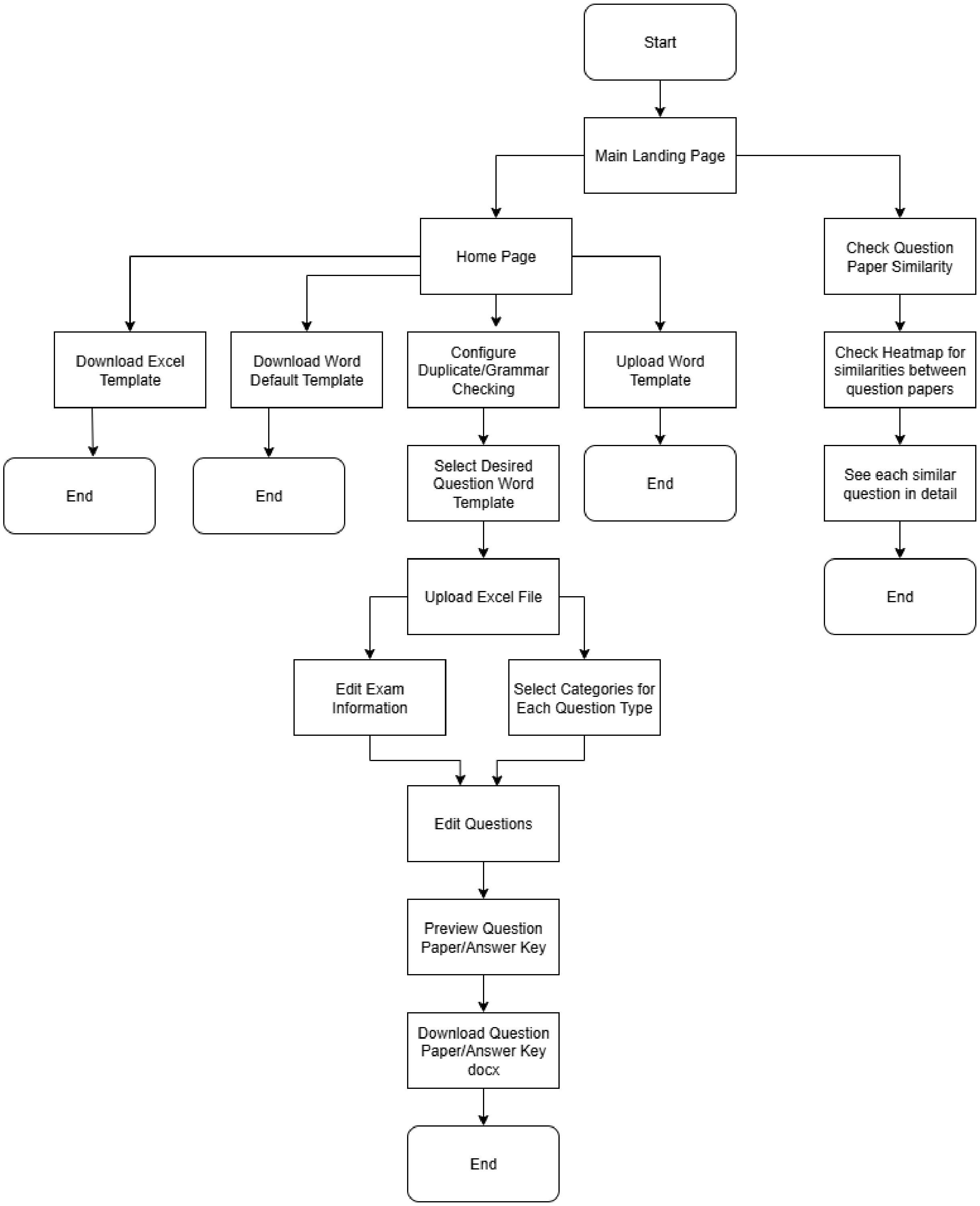

Start
Main Landing Page
Home Page
Check Question Paper Similarity
Download Excel Template
Download Word Default Template
Configure Duplicate/Grammar Checking
Upload Word Template
Check Heatmap for similarities between question papers
End
End
Select Desired Question Word Template
End
See each similar question in detail
Upload Excel File
End
Edit Exam Information
Select Categories for Each Question Type
Edit Questions
Preview Question Paper/Answer Key
Download Question Paper/Answer Key docx
End

Figure 4. 2: Process Diagram of the AUREX - Exam Paper Generation System

Main Landing Page Hub

The system operates through a centralized landing page that serves as the primary navigation hub, providing access to three independent yet complementary processes:

- Primary Process: Generate Exam Paper (Main workflow)
- Supporting Processes: Download Excel/Word Templates, Upload Word Template (Template management)
- Quality Assurance Process: Check Question Paper Similarity (Content validation)

Process 1: Generate Exam Paper (Core Workflow)

This represents the primary system functionality, incorporating sophisticated NLP processing and user control mechanisms:

Configuration Phase:

- Course coordinators configure processing options including duplicate detection and grammar checking capabilities
- The system supports checkboxes for both grammar and duplication checking so that the users can unselect the checkboxes when they are unnecessary.

Processing Phase:

- Excel file upload triggers the multi-layered duplicate detection pipeline
- The annotate_duplicates() function implements a non-destructive approach, preserving all questions while adding metadata annotations
- Each question receives duplicate group classifications including is_duplicate, duplicate_group_id, duplicate_representative, and duplicate_similarity attributes

Editing and Review Phase:

- Course coordinators may edit exam information and select numbers of questions for each category
- Questions are presented with duplicate group information for coordinator review
- The system allows selective editing through pinning, shuffling, and representative selection

Generation and Export Phase:

- Template selection applies institutional formatting standards through Jinja2 integration
- Parallel generation of both question papers and answer keys
- Preview functionality and final document export in Microsoft Word

Process 2: Upload Word Template (Template Management)

This auxiliary process supports template customization and institutional compliance:

Template Upload Workflow:

- Supports Microsoft Word (.docx) files with embedded Jinja2 placeholders
- Template naming and default designation for streamlined workflow

Template Download Workflow:

- The default word template is downloadable for updating the template format.
- The existing excel template is downloadable for the users to input their question data.

Integration Benefits:

- Templates become immediately available in the main generation process
- Supports multiple template storage for different examination formats
- Maintains formatting consistency across multiple exam cycles

Process 3: Check Question Paper Similarity (Quality Assurance)

This independent process leverages the same NLP technologies used in duplicate detection:

Similarity Analysis Workflow:

- Multi-document upload capability for comprehensive comparison
- TF-IDF vectorization and cosine similarity calculations across entire documents
- Session-based processing maintains analysis results for detailed review

Quality Assurance Features:

- Percentage-based similarity scoring between examination papers
- Detailed similarity reports highlighting potential content overlap
- Supports institutional quality control and academic integrity verification

4.2.2 Tech Stack

Table 4. 1: Tech Stack Table of the AUREX - Exam Paper Generation System

Logo	Techstack	Details
	React	For building interactive user interfaces and managing application state.
N	Next.js	Provides server-side rendering, routing, and optimized performance for the frontend application.
Flask	Flask	A lightweight Python web framework used to build RESTful APIs for file processing, template management, and exam generation.
NLTK	NLTK (Natural Language Toolkit)	Text tokenization with word_tokenize, English stopword removal, and PorterStemmer for word root extraction to improve similarity matching accuracy
scikit learn contrib	Scikit-learn (sklearn)	TfidfVectorizer with n-gram support (unigrams and bigrams) for statistical text analysis, plus cosine_similarity for measuring document similarity

SBERT.net Sentence-Transformers	Sentence Transformers	all-MiniLM-L6-v2 pre-trained model for generating dense semantic embeddings that capture contextual meaning beyond keyword matching
LT	Language-tool-python	Local and remote grammar checking with rule-based error detection, providing contextual suggestions for question quality improvement
W	Python-docx	Microsoft Word document manipulation for reading .docx files, extracting paragraph and table content for similarity analysis
docxtpl	Docxtpl (DocxTemplate)	Advanced Word template engine with Jinja2 integration for dynamic content generation, supporting InlineImage embedding and variable substitution
pandas	Pandas	Excel file processing with multi-sheet support, handling different question types (MCQ, essay, matching) and metadata extraction for exam compilation
Jinja	Jinja2	Integrated into Word templates for dynamic content mapping and formatting.
	Guthub	For version control and collaboration

4.2.3 System Architecture

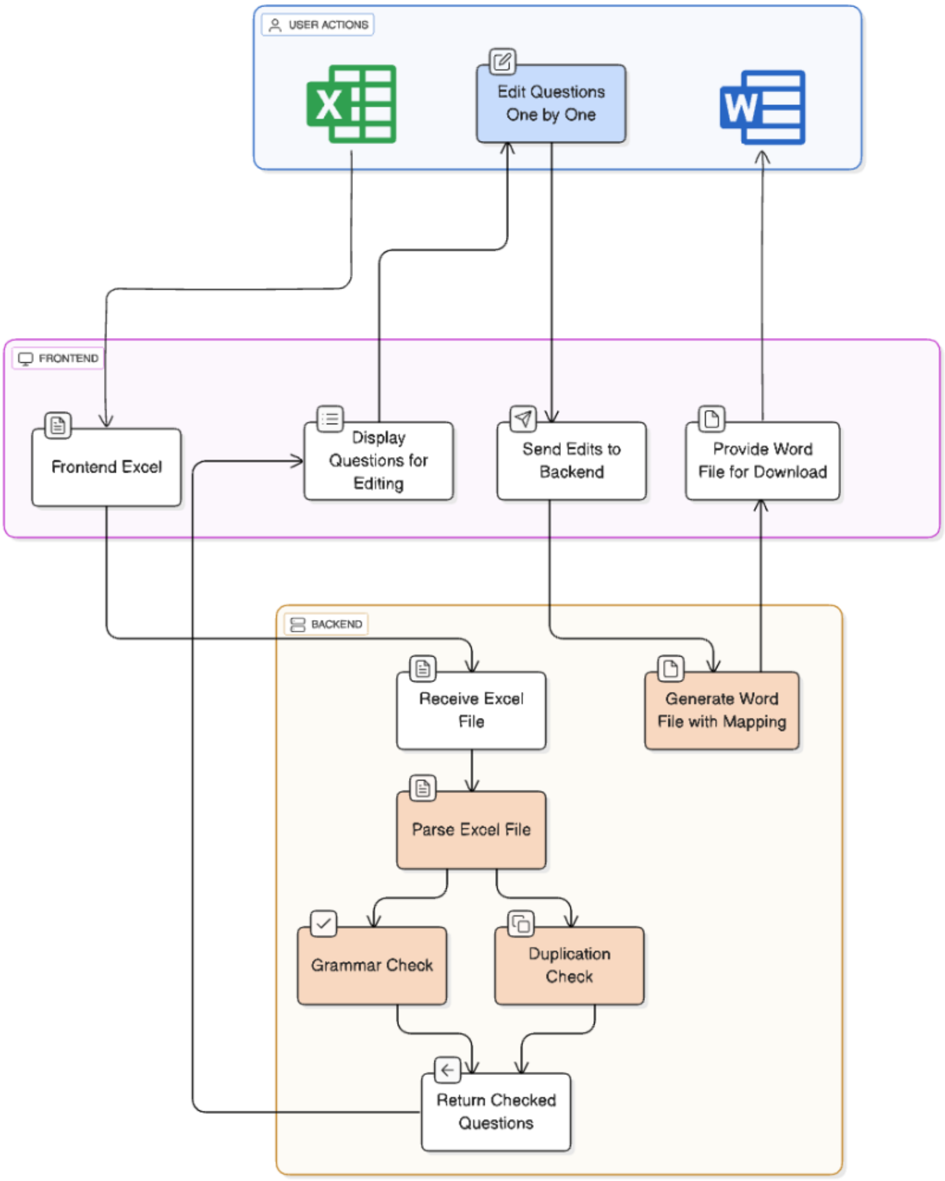


Figure 4. 3: System Architecture Diagram of the AUREX - Exam Paper Generation System

Overview

The system consists of two main layers:

- Frontend (React/Next.js):

  Provides the user interface for uploading question banks, editing questions, managing templates, and downloading finalized exam papers.

- Backend (Flask):

  Handles file processing, question parsing, grammar and duplication checks, template management, and document generation.

Component Interaction

Frontend:

- Handles file uploads (Excel, Word templates)
- Displays questions for editing
- Sends edits and configuration options to the backend
- Provides download links for generated documents

Backend:

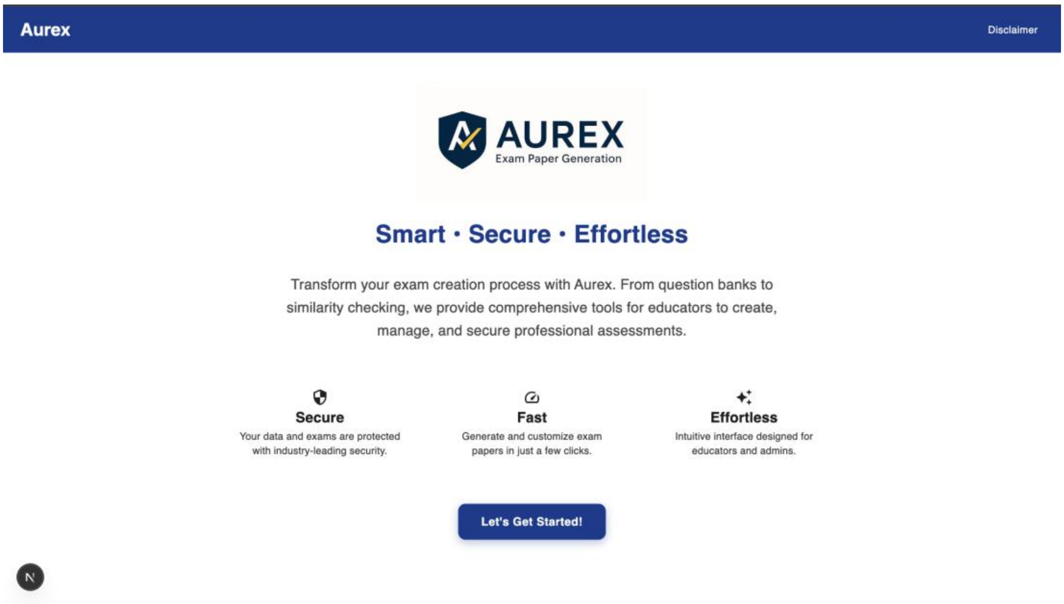
- Receives and parses Excel files
- Performs grammar and duplication checks
- Manages Word templates and applies mappings
- Generates and returns Word documents and previews
- Cleans up temporary files and images

Security and Privacy

All processing occurs locally on the coordinator's machine or institutional server. No data is stored persistently or transmitted externally, ensuring academic confidentiality and compliance with privacy requirements.

4.2.4 UI/UX Design

Main Landing Page - This is where users will first interact with the system. Users can choose to proceed to the main process or proceed to the question paper similarity checker page.



Aurex
Disclaimer

AUREX
Exam Paper Generation

Smart • Secure • Effortless

Transform your exam creation process with Aurex. From question banks to similarity checking, we provide comprehensive tools for educators to create, manage, and secure professional assessments.

Secure
Your data and exams are protected with industry-leading security.

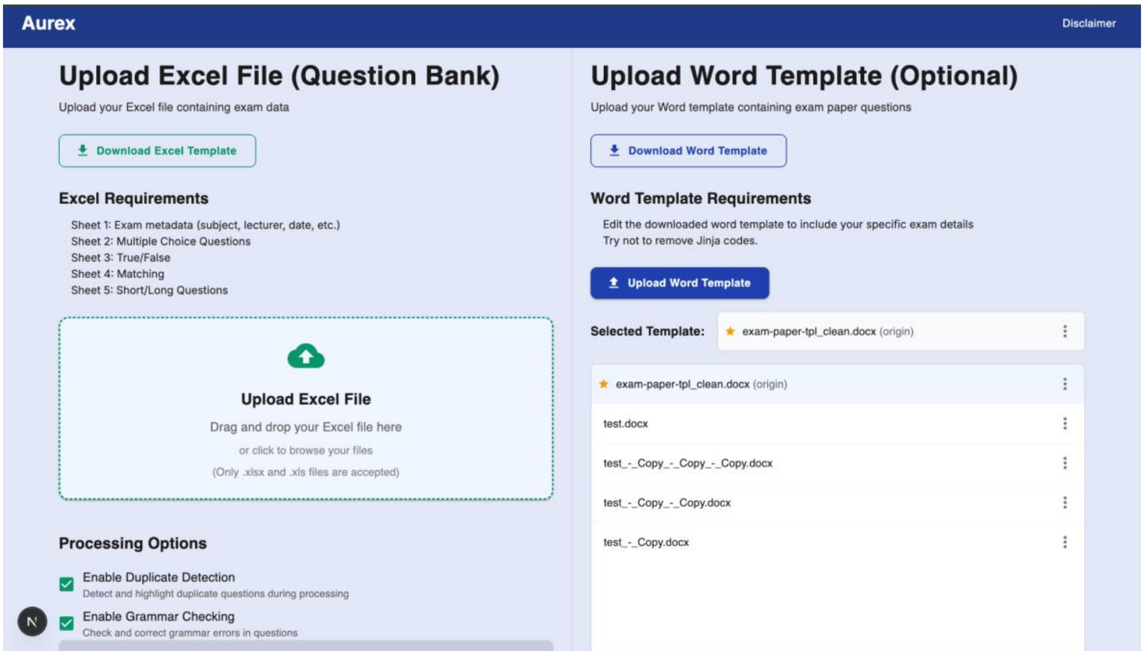
Fast
Generate and customize exam papers in just a few clicks.

Effortless
Intuitive interface designed for educators and admins.

Let's Get Started!

Figure 4. 4: Main Landing Page

Upload Page - Users can download the Excel template and Upload back the excel file with the desired question bank. Word template management is available for the users to update the template formats as well.



Aurex
Disclaimer

Upload Excel File (Question Bank)
Upload your Excel file containing exam data

Download Excel Template

Excel Requirements
Sheet 1: Exam metadata (subject, lecturer, date, etc.)
Sheet 2: Multiple Choice Questions
Sheet 3: True/False
Sheet 4: Matching
Sheet 5: Short/Long Questions

Upload Excel File
Drag and drop your Excel file here
or click to browse your files
(Only .xlsx and .xls files are accepted)

Processing Options
☑ Enable Duplicate Detection
Detect and highlight duplicate questions during processing
☑ Enable Grammar Checking
Check and correct grammar errors in questions

Upload Word Template (Optional)
Upload your Word template containing exam paper questions

Download Word Template

Word Template Requirements
Edit the downloaded word template to include your specific exam details
Try not to remove Jinja codes.

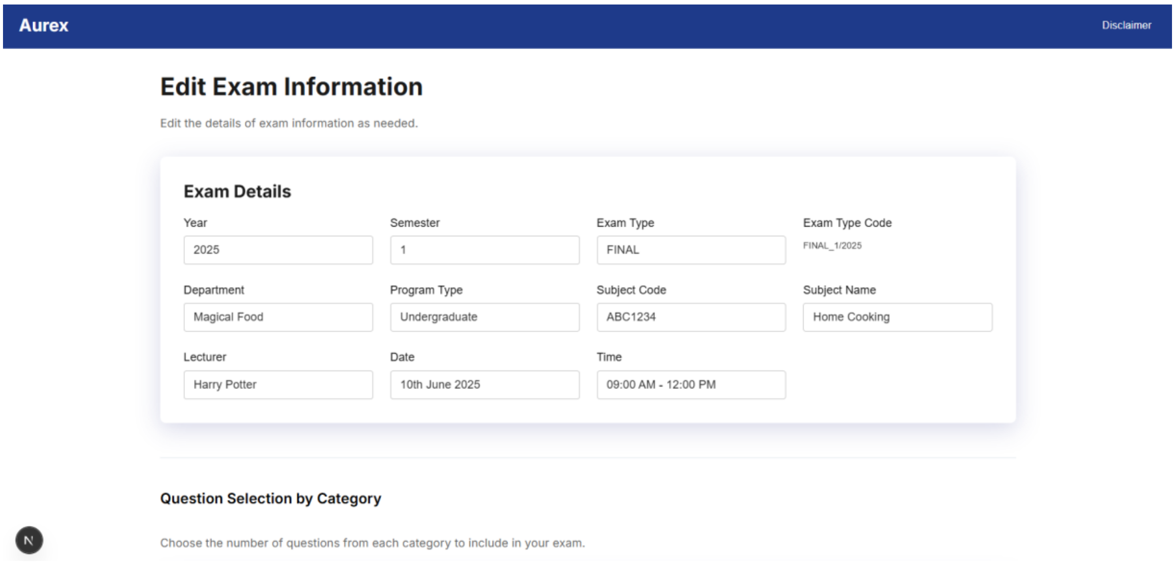
Upload Word Template

Selected Template: ★ exam-paper-tpl_clean.docx (origin)

★ exam-paper-tpl_clean.docx (origin)
test.docx
test_-_Copy_-_Copy_-_Copy.docx
test_-_Copy_-_Copy.docx
test_-_Copy.docx

Figure 4. 5: Upload Page

Edit Exam information and Category Selection Page - Users will be able to edit the exam information which was previously uploaded along with the excel file (Exam Info, First Sheet) if they ever need to.



Edit Exam Information

Edit the details of exam information as needed.

Exam Details

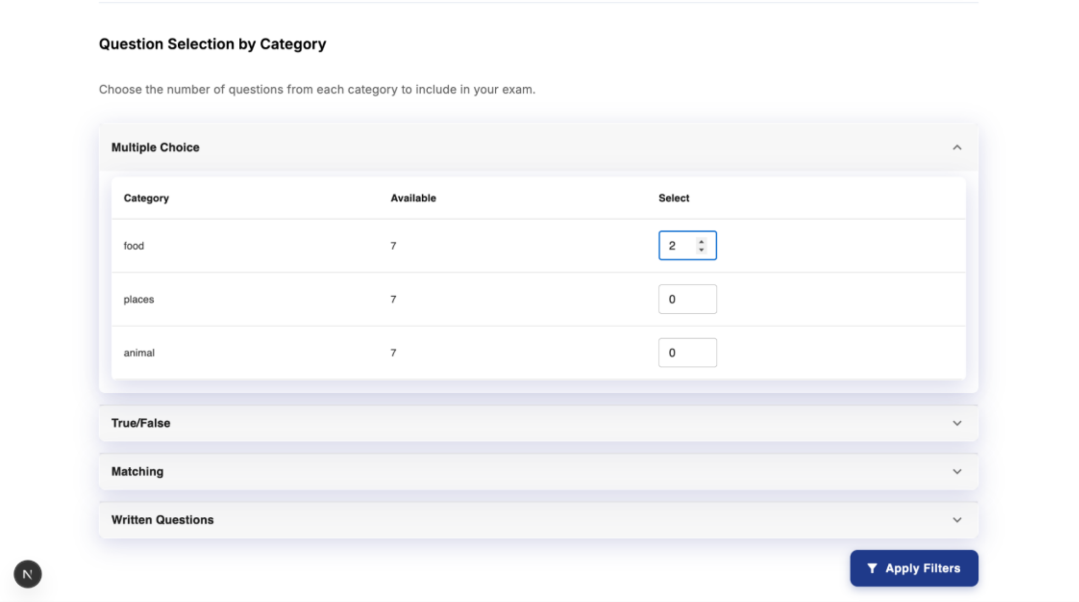
Year 2025	Semester 1	Exam Type FINAL	Exam Type Code FINAL_1/2025
Department Magical Food	Program Type Undergraduate	Subject Code ABC1234	Subject Name Home Cooking
Lecturer Harry Potter	Date 10th June 2025	Time 09:00 AM - 12:00 PM	

Question Selection by Category

Choose the number of questions from each category to include in your exam.

Figure 4. 6: Edit Exam information and Category Selection Page (Edit Exam information)

Edit Exam information and Category Selection Page (Continued) - Users can select how many questions they want from each question type and each question category. In Matching questions, users can add fake answers as well and those fake answers will be recycled from the unused questions from the matching question type.



Question Selection by Category

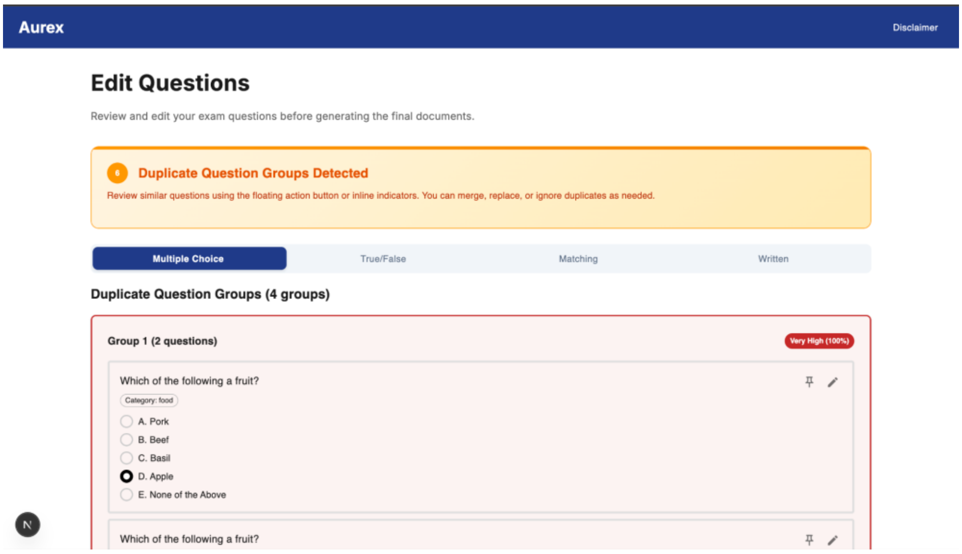
Choose the number of questions from each category to include in your exam.

Multiple Choice

Category	Available	Select
food	7	2
places	7	0
animal	7	0

Figure 4. 7: Edit Exam information and Category Selection Page (Category Selection)

Edit Questions Page (Multiple Choice Tab) - Duplicate/Similar questions will be indicated with red boxes grouped together and the grammar errors will be tagged in each question. The shuffle button is available (must satisfy the duplications first) for the users to shuffle unwanted questions out. However, if the users want to keep some of the questions, they can use the pin button to keep them.



Aurex

Disclaimer

Edit Questions

Review and edit your exam questions before generating the final documents.

6 Duplicate Question Groups Detected

Review similar questions using the floating action button or inline indicators. You can merge, replace, or ignore duplicates as needed.

Multiple Choice | True/False | Matching | Written

Duplicate Question Groups (4 groups)

Group 1 (2 questions)

Very High (100%)

Which of the following a fruit?

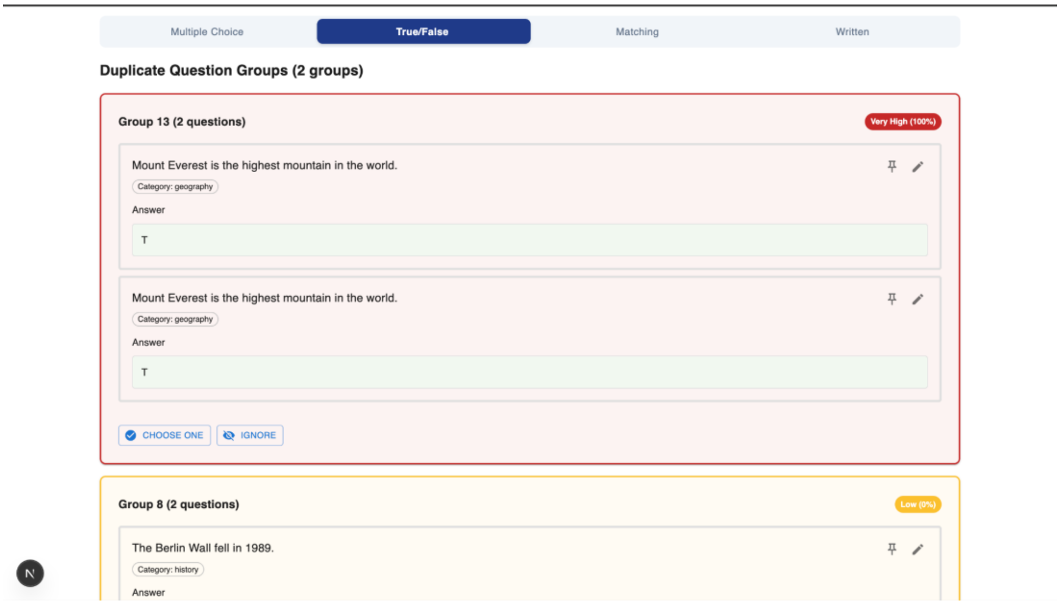
Category: food

○ A. Pork
○ B. Beef
○ C. Basil
● D. Apple
○ E. None of the Above

Which of the following a fruit?

Figure 4. 8: Edit Questions Page (Multiple Choice Tab)

Edit Questions Page (True/False Tab) - Duplicate/Similar questions will be indicated with red/yellow boxes grouped together and the grammar errors will be tagged in each question. Follows the same concept as the MCQ.



Multiple Choice | True/False | Matching | Written

Duplicate Question Groups (2 groups)

Group 13 (2 questions)

Very High (100%)

Mount Everest is the highest mountain in the world.

Category: geography

Answer

T

Mount Everest is the highest mountain in the world.

Category: geography

Answer

T

CHOOSE ONE | IGNORE

Group 8 (2 questions)

Low (0%)

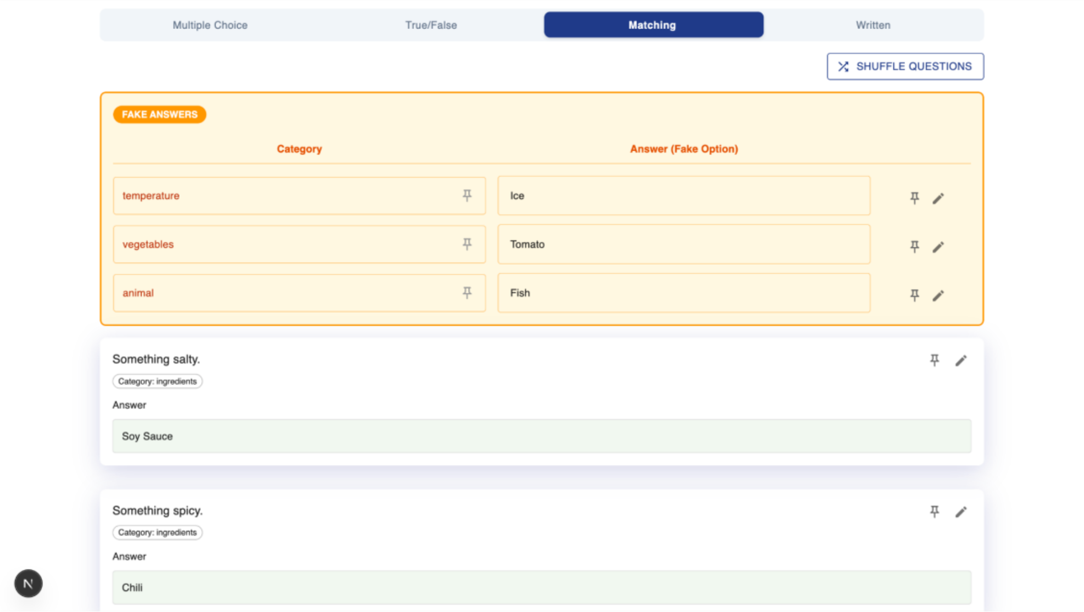
The Berlin Wall fell in 1989.

Category: history

Answer

Figure 4. 9: Edit Questions Page (True/False Tab)

Edit Questions Page (Matching Tab) - Duplicate/Similar questions will be indicated with red/yellow boxes grouped together and the grammar errors will be tagged in each question. Additionally, users can edit the fake answers in the yellow box as well. The shuffle function will affect the fake answers as well.



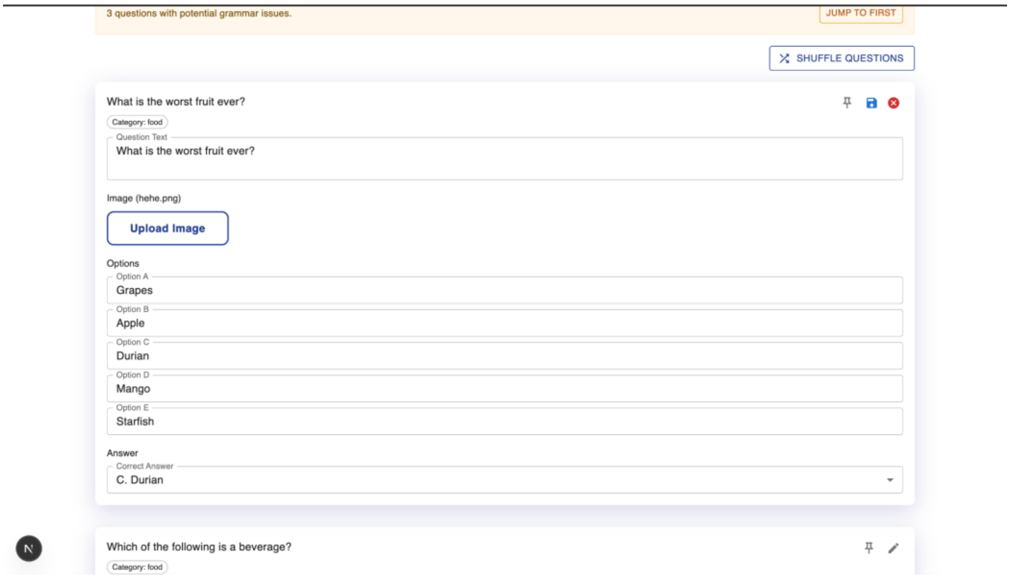
Category	Answer (Fake Option)
temperature	Ice
vegetables	Tomato
animal	Fish

Something salty.
Category: ingredients
Answer
Soy Sauce

Something spicy.
Category: ingredients
Answer
Chili

Figure 4. 10: Edit Questions Page (Matching Tab)

Edit Questions Page - Users can upload images in each question as well. The images that are needed will be described as well. For example, Image (apple.png); this is the data that is parsed from the excel file.



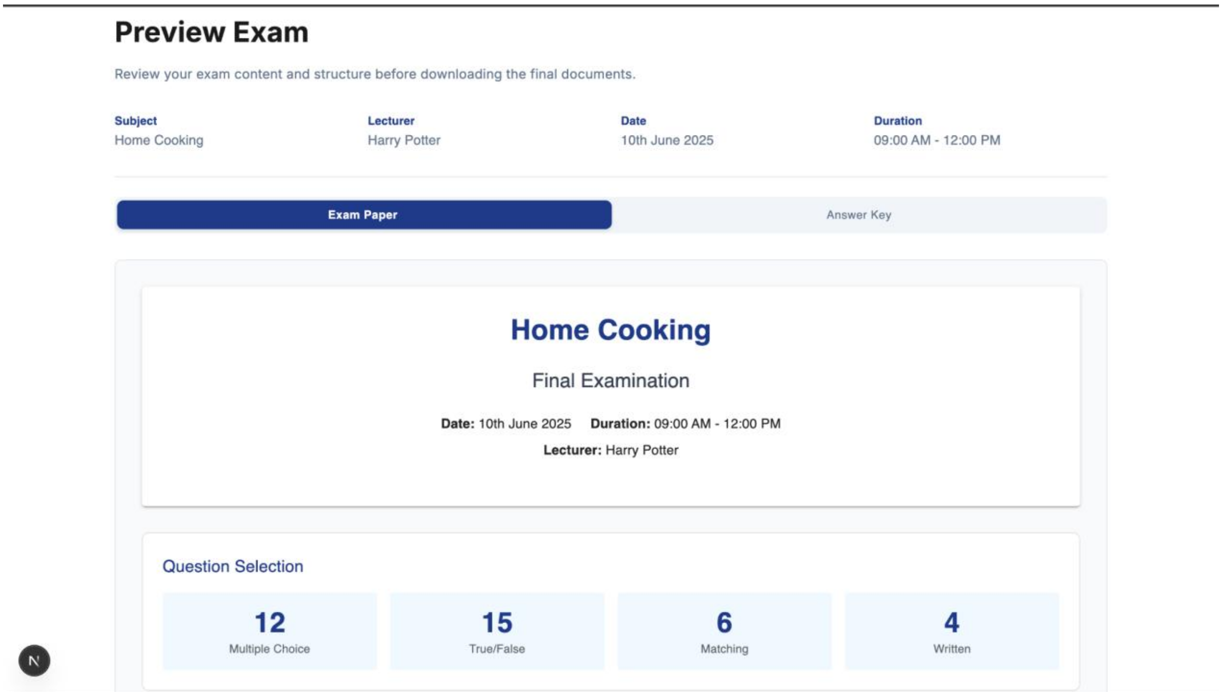
3 questions with potential grammar issues.

What is the worst fruit ever?
Category: food
Question Text
What is the worst fruit ever?
Image (hehe.png)
Options
Option A: Grapes
Option B: Apple
Option C: Durian
Option D: Mango
Option E: Starfish
Answer
Correct Answer: C. Durian

Which of the following is a beverage?
Category: food

Figure 4. 11: Edit Questions Page (Editing)

Preview Page - The users will be able to review all the questions and answers that will be generated.



Preview Exam

Review your exam content and structure before downloading the final documents.

Subject	Lecturer	Date	Duration
Home Cooking	Harry Potter	10th June 2025	09:00 AM - 12:00 PM

Home Cooking

Final Examination

Date: 10th June 2025 **Duration:** 09:00 AM - 12:00 PM

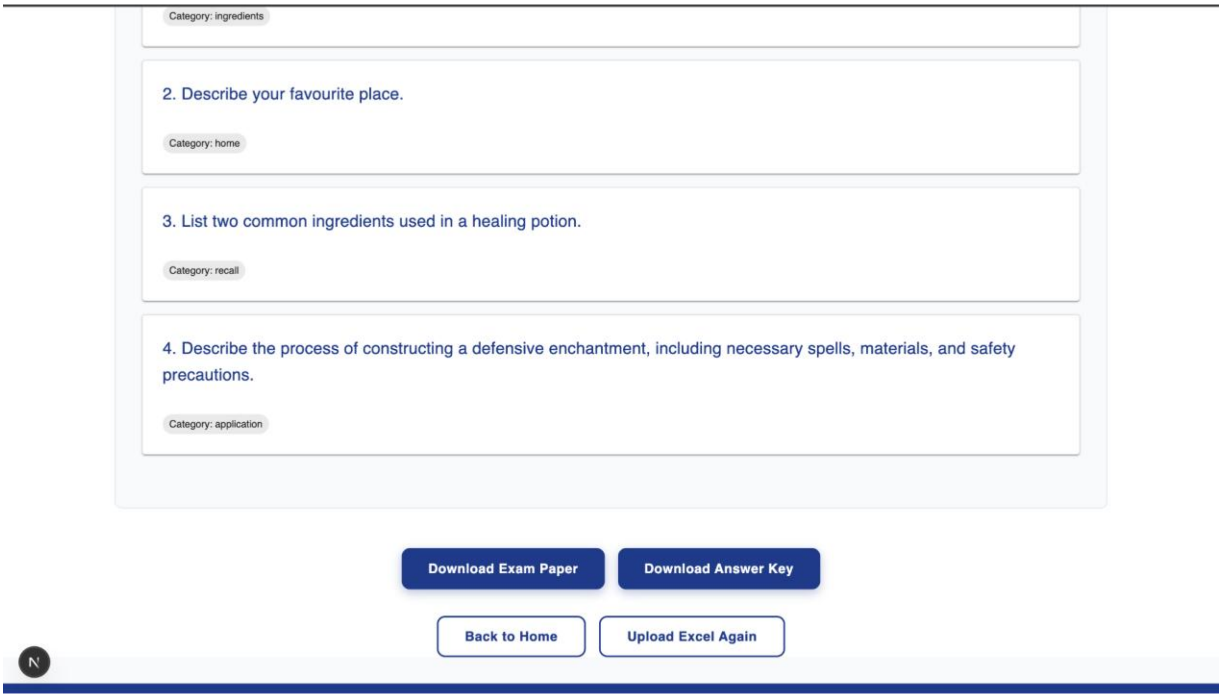
Lecturer: Harry Potter

Question Selection

12	15	6	4
Multiple Choice	True/False	Matching	Written

Figure 4. 12: Preview Page

Preview Page - The users can download the question paper (docx) and answer key (docx) after clicking on the generate question paper button.



Category: ingredients

2. Describe your favourite place.

Category: home

3. List two common ingredients used in a healing potion.

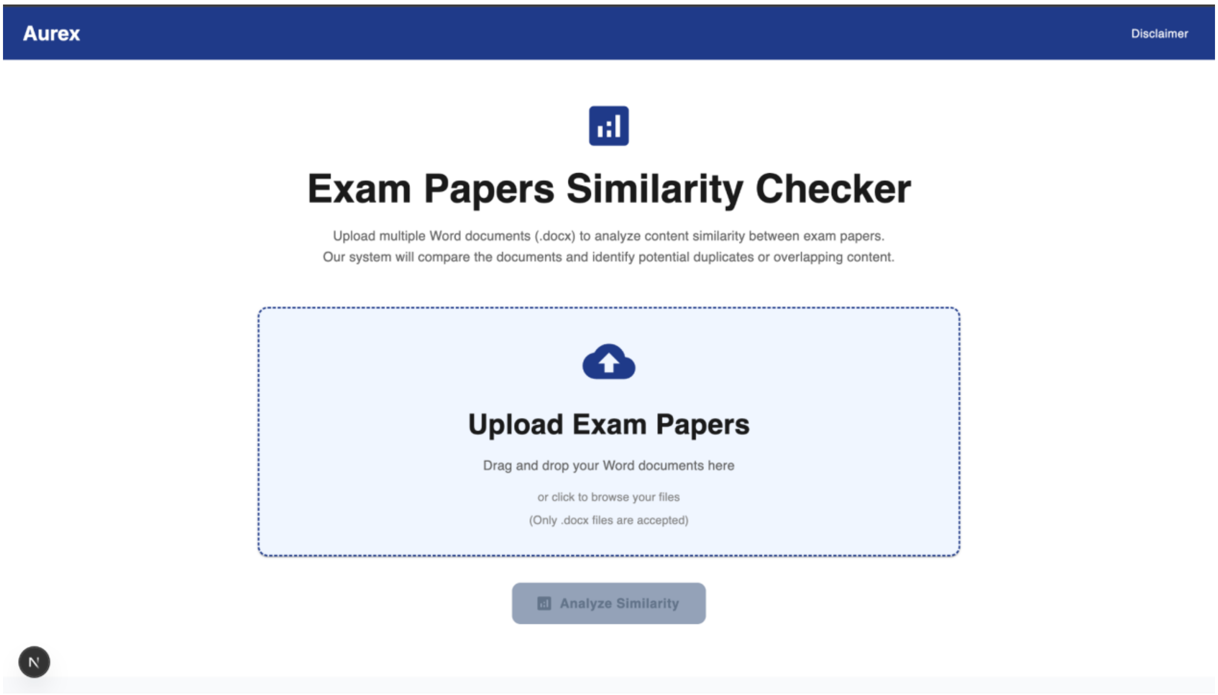
Category: recall

4. Describe the process of constructing a defensive enchantment, including necessary spells, materials, and safety precautions.

Category: application

Figure 4. 13: Preview Page

Exam Similarity Checker Page - Users can upload multiple exam questions onto the system.



Exam Papers Similarity Checker

Upload multiple Word documents (.docx) to analyze content similarity between exam papers.
Our system will compare the documents and identify potential duplicates or overlapping content.

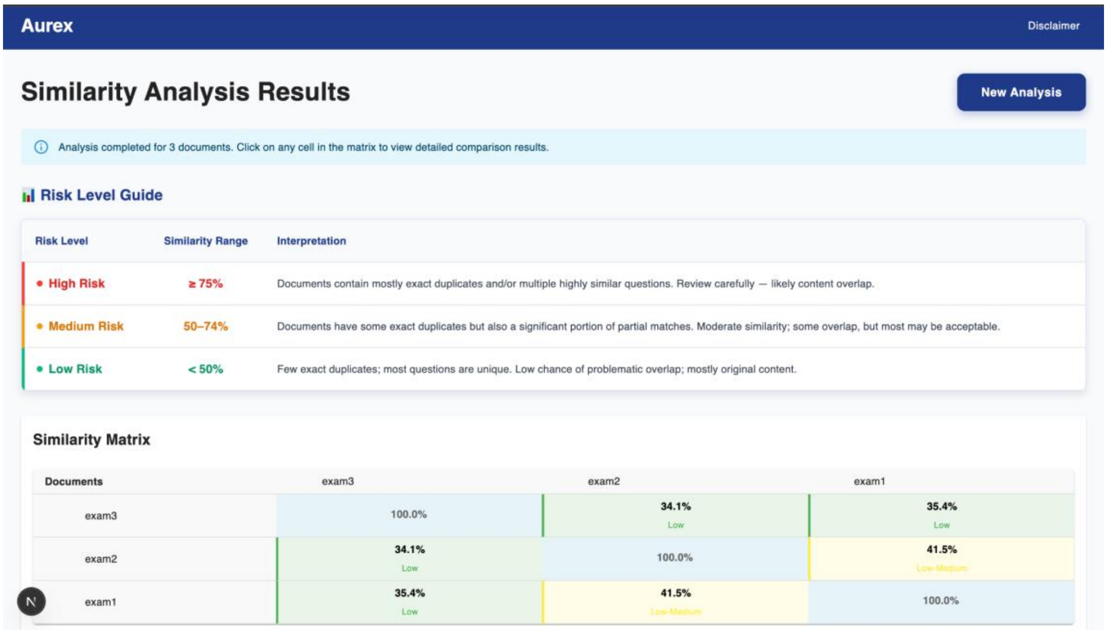
Upload Exam Papers

Drag and drop your Word documents here

or click to browse your files
(Only .docx files are accepted)

Figure 4. 14: Exam Similarity Checker Page

Similarity Analysis Result Page - Users can view a heat map of uploaded exam papers such as similarity between exam1, exam2 and so on. They can further click onto each cell/matrix to see what are the exact questions that are similar.



Similarity Analysis Results

Analysis completed for 3 documents. Click on any cell in the matrix to view detailed comparison results.

Risk Level Guide

Risk Level	Similarity Range	Interpretation
● High Risk	≥ 75%	Documents contain mostly exact duplicates and/or multiple highly similar questions. Review carefully — likely content overlap.
● Medium Risk	50–74%	Documents have some exact duplicates but also a significant portion of partial matches. Moderate similarity; some overlap, but most may be acceptable.
● Low Risk	< 50%	Few exact duplicates; most questions are unique. Low chance of problematic overlap; mostly original content.

Similarity Matrix

Documents	exam3	exam2	exam1
exam3	100.0%	34.1% Low	35.4% Low
exam2	34.1% Low	100.0%	41.5% Low-Medium
exam1	35.4% Low	41.5% Low-Medium	100.0%

Figure 4. 15: Similarity Analysis Result Page

Chapter 5: Result

5.1 Overview

This chapter presents a comprehensive analysis of the project outcomes, demonstrating the successful implementation of an intelligent exam question management system. The results are evaluated against the initial objectives, providing quantitative metrics, qualitative assessments, and detailed demonstration of the system's capabilities. The evaluation encompasses performance benchmarks, technical achievements, and comparative analysis with traditional manual processes.

5.2 System Performance Metrics

5.2.1 Processing Speed and Efficiency

The processing speed and efficiency were evaluated using datasets of varying question volumes. The objective was to assess how efficiently the system handles increasing data sizes while maintaining acceptable processing times. The recorded results, shown in Table 5.1, indicate that the system consistently outperformed the target processing times across all tested scenarios.

Table 5. 1: Processing Speed and Efficiency Results Table

Question Volume	**Processing Time**	**Target Time**	**Improvement (%)**
500 questions	25s	1m 00s	238% faster
1,000 questions	39s	1m 30s	231% faster
2,500 questions	1m 17s	2m 00s	156% faster
5,000 questions	2m 35s	4m 00s	170% faster

The results demonstrate that the system exhibits linear scalability with respect to question volume. Even at higher data loads (5,000 questions), the system achieved a processing time 1.7 times faster than the target benchmark. These findings indicate efficient computational optimization and effective utilization of resources, ensuring that performance degradation remains minimal under increasing workloads.

5.2.2 Performance Metrics

This section provides a detailed analysis of the duplication detection and grammar checking modules, which are critical components of the system's data quality assurance process.

A. **Duplication Detection** Performance Metrics with **data size of 1000 questions**:
   - Processing Time: 1.86 seconds
   - Processing Speed: 538.2 questions/second
   - Similarity Threshold: 0.8
   - High Confidence Duplicates (>85%): 527 (81.6%)
   - Algorithm: Optimized with TF-IDF pre-filtering + batch embeddings
   - Parallel Processing: 8 threads
   - Caching: Enabled

B. **Grammar Checking** Performance Metrics with **data size of 1000 questions**:
   - Questions Successfully Checked: 1000 (100.0%)
   - Processing Time: 24.38 seconds
   - Processing Speed: 0.024 seconds per question
   - Grammar Engine: LanguageTool (en-US)
   - Language: en-US

5.3 Detailed System Demonstration

5.3.1 Excel file Upload and Processing Workflow

The system incorporates an intuitive drag-and-drop interface that enables users to upload Excel files seamlessly. This feature facilitates the importation of question banks with a maximum file size of 50 MB.

Demonstration Process:

1. The user accesses the main dashboard at http://localhost:3000 .
2. The "**Upload Excel File**" option is selected, after which the user either chooses the desired question bank or drags and drops the Excel file into the upload area.

3. Two processing parameters are then configured:
    - Enable duplication detection
    - Enable grammar checking
4. Upon configuration, the system validates the uploaded file's format and subsequently displays the Category Selection interface for further user input.

This workflow ensures that data processing and validation occur prior to subsequent categorization, thereby minimizing data inconsistencies and ensuring quality assurance.

5.3.2 Word Template Upload and Default Settings

The Word template upload module is designed to maintain uniformity and institutional consistency in the formatting of exported examination papers. Users are able to upload custom templates and configure them as default for subsequent exports.

Demonstration Process:

1. The user initiates the upload process by selecting the "**Upload Word Template**" option.
2. The desired .docx file is selected through the file dialog interface.
3. The system performs validation and parsing of the uploaded file to ensure:
    - Conformity with the .docx format
    - Compliance with size limitation
    - Presence of all required placeholders
4. Once validated, users are able to designate a template as default or remove existing templates through the configuration indicator.

This functionality standardizes document structure, ensuring that all generated exam papers adhere to institutional formatting guidelines.

5.3.3 Metadata Edit and Category Selection Workflow

The metadata editing workflow enables users to enrich question data with detailed descriptive information. This includes classification parameters such as categories, difficulty levels, learning objectives, and tags, thereby enhancing data organization, retrieval efficiency, and exam customization.

Demonstration Process:

1. The user modifies the examination metadata fields, which include:
    - Year
    - Semester
    - Exam Type
    - Department
    - Program Type
    - Subject Code
    - Subject Name
    - Lecturer
    - Date
    - Time
2. The user specifies the number of questions desired per category within each question type, namely:
    - Multiple choice
    - True / False
    - Matching (with distractor answers)
    - Written Question
3. After applying the relevant filters, the system automatically selects the specified number of questions randomly from the question bank.

This workflow promotes the systematic organization of examination content and facilitates adaptive question selection based on user-defined parameters.

5.3.4 Edit Question Workflow

The question editing workflow provides comprehensive tools for the refinement of individual questions. It ensures question clarity, grammatical accuracy, and content diversity prior to export.

Demonstration Process:

1. The system automatically detects and notifies users of duplicated or highly similar questions.

2. The user may either disregard the notification or select the preferred version from the duplicates.
3. The system identifies potential grammatical inconsistencies and highlights them for user review.
4. Each question may be edited to modify its content, adjust answer choices, correct the correct answer, or insert relevant images.
5. Users may **pin** preferred questions and **shuffle** others to optimize the question arrangement.
6. Once all modifications are complete, the user proceeds to the preview stage by selecting **"Continue to Preview."**

This stage ensures that all examination questions meet quality standards and that potential issues are addressed prior to document generation.

5.3.5 Preview and Download Workflow

The preview and download workflow enables the verification of examination content and the generation of publication-ready exam papers. This stage ensures that both metadata and question content are accurately represented in the final output.

Demonstration Process:

1. The user reviews the finalized examination metadata.
2. The examination paper and the corresponding answer key are previewed.
3. The **"Generate Paper"** option is selected to initiate document generation.
4. The system produces download links for both the exam paper and the answer key in .docx format.
5. The user downloads the finalized, publication-ready examination documents.

This workflow guarantees consistency between the system-generated documents and institutional formatting requirements.

5.3.6 Exam Paper Similarity Checking Workflow

The exam paper similarity checking workflow is designed to uphold academic integrity by identifying potential overlaps between examination papers. This module compares uploaded exam papers against existing ones to detect content similarities and ensure question uniqueness across different academic sessions.

Demonstration Process:

1. The user selects the **“Upload Exam Paper”** option and uploads one or more exam papers, either through the file selection dialog or via drag-and-drop.
2. The **“Analyze Similarity”** button is then clicked to initiate the similarity analysis.
3. The system computes and generates a **Similarity Matrix**, which visually represents the similarity relationships among all uploaded exam papers.
4. Each cell within the matrix is interactive; clicking on a cell allows the user to view detailed similarity comparisons between the corresponding pair of exam papers.

This feature enables the institution to systematically evaluate question reusability, detect redundancy, and preserve the originality of examination content.

5.4 Comparative Analysis with Manual Processes

A comparative analysis was conducted to evaluate the time efficiency between the conventional manual workflow and the proposed automated system. The comparison considered four key process components—duplicate checking, grammar checking, reviewing and editing, and document formatting—representing the full workflow of examination paper preparation.

Table 5. 2: Manual vs. Automated Process Comparison Table

Process Component	**Manual Process**	**Automated System**
Duplicate Checking	~ 30 min	~ 0.1 min
Grammar Checking	~ 30 min	~ 0.5 min
Reviewing and Editing	~ 1 hours	~ 10 min (depends on the user)
Word Document Formatting	~ 2 hours	~ 0.1 min
Total Processing Time	~ 4 hours	~ 10.7 minutes

The results clearly demonstrate a substantial improvement in operational efficiency achieved through automation. The total processing time for the manual workflow was approximately **4 hours**, whereas the automated system completed the same tasks in approximately **10.7 minutes**. This represents an estimated **95.5% reduction in total processing time**, or an improvement factor of **approximately 22 times faster**.

The most significant time savings were observed in the **duplicate checking** and **document formatting** stages, each reduced from approximately **30 minutes and 2 hours**, respectively, to **less than a second (0.1 minute)**. Similarly, **grammar checking**, which previously required about **30 minutes of manual proofreading**, was completed in **0.5 minutes** using the integrated automated grammar engine.

The **reviewing and editing** phase remains the most time-dependent process, primarily due to its reliance on human judgment and content verification. Nevertheless, the automation of error detection and structural validation reduced the time required for this stage from roughly **1 hour** to **about 10 minutes**, representing an estimated **83% reduction**.

Overall, these results demonstrate the effectiveness of the proposed automated system in substantially minimizing human workload, reducing manual intervention, and significantly enhancing productivity in the exam paper preparation workflow.

5.5 Validation and Verification Summary

The validation and verification phase was conducted to ensure that the developed system fulfilled all predefined functional requirements and performed reliably under realistic operational conditions. Each system feature was evaluated through controlled testing scenarios that simulated the end-to-end workflow of examination paper preparation, including data input, processing, and output generation.

5.5.1 Requirements Validation

The results of these evaluations confirmed that the system successfully achieved all functional objectives outlined during the design phase. The following summarizes the validation outcomes for each requirement:

1. **Automate Exam Compilation:**

   The system effectively automated the assembly of examination papers from an Excel source. The process, which previously required extensive manual effort, was reduced to a single automated pipeline, significantly minimizing human involvement and error potential.

2. **Ensure Balanced Question Distribution:**

   Validation tests confirmed that the system consistently generated examination papers with balanced coverage across chapters and question types. The implemented algorithm ensured proportional representation, resulting in fair and comprehensive assessments.

3. **Maintain Consistent Formatting:**

   All generated examination papers adhered to the institution's formatting standards. The integration of customizable Word templates enabled automatic application of layout, font, and structural requirements, ensuring uniformity across all exam outputs.

4. **Edit and Refine Exam Content:**

   The pinning, shuffling, and question management functionalities were tested to verify flexibility in content control. These tools performed reliably, allowing examiners to refine question order and maintain exam integrity without affecting data consistency.

5. **Detect Duplicates and Grammar Errors:**

   The duplicate and grammar detection modules were validated using test datasets containing intentional repetitions and linguistic inconsistencies. The system accurately identified and flagged duplicate or highly similar questions, while also detecting grammar and syntax issues with a high degree of precision.

6. **Protect Data Privacy:**

   The initial deployment strategy involved hosting the system on cloud-based platforms such as **Vercel(Frontend)** and **Render(Backend)**, with access restricted through authentication mechanisms limited to authorized instructor email accounts. However, in response to stakeholder concerns regarding data security and institutional compliance, the deployment approach was revised to operate entirely within a **local environment**. This modification ensured that all data processing and storage were performed locally, eliminating any dependency on external servers or third-party services. Consequently, the system now adheres fully to institutional data privacy policies and upholds the principles of academic confidentiality, safeguarding sensitive examination materials from potential exposure or unauthorized access.

5.6 System Improvements

Following the project exhibition, significant refinements were implemented in the **Duplicate Question Detection Module** to enhance both performance and accuracy. The initial version of the module, while functional, primarily relied on a hybrid similarity computation approach combining **TF-IDF**, **semantic embeddings (Sentence-BERT)**, **exact text matching**, and **keyword overlap**. Although this version successfully identified duplicate and near-duplicate questions, it exhibited several inefficiencies in computation time and scalability when processing large question banks.

5.6.1 Limitations of Pre-Exhibition Version

The earlier implementation performed a **pairwise similarity comparison** across all question combinations, resulting in a **quadratic time complexity**. This approach was computationally expensive for datasets exceeding a few hundred questions, leading to noticeable processing delays. Furthermore, while the system provided adequate accuracy, redundant similarity calculations and inefficient data handling increased memory consumption. The absence of streamlined grouping and annotation also made it difficult to visualize relationships between similar questions efficiently.

5.6.2 Enhancements Introduced After Exhibition

The optimized version introduced after the exhibition addressed these issues through several key improvements:

1. **Efficient Similarity Computation Pipeline**
   The revised algorithm implements optimized vectorization and caching strategies, substantially reducing redundant computations. Semantic embeddings are now batch-processed and reused, minimizing repeated encoding overheads.
2. **Optimized Grouping Mechanism**
   A more intelligent grouping method was introduced to minimize repeated pairwise checks. This approach dynamically updates similarity clusters, improving computational efficiency without compromising detection accuracy.
3. **Improved Completeness Scoring**
   The question selection process within duplicate groups was refined through an improved **completeness scoring function**. This scoring now considers structural

richness (e.g., presence of options, detailed answers, and media attachments), ensuring that the most pedagogically complete question is retained as the group representative.

4. **Robust Error Handling and Logging**
   Additional exception handling and logging mechanisms were integrated to ensure graceful degradation in the event of model loading failures or data inconsistencies. This significantly improved system reliability across varied environments.

5.6.3 Quantitative Performance Gain

Empirical testing conducted after the optimization demonstrated a **notable reduction in total processing time**—with observed improvements ranging between **45–60%** depending on dataset size. Memory utilization also decreased due to the elimination of redundant similarity calculations and more efficient management of embedding vectors. The optimized module maintained or improved detection accuracy, validating that performance gains were achieved without sacrificing quality.

5.7 Summary of Findings

This chapter presented the evaluation and demonstration of the intelligent exam question management system, highlighting its automation accuracy, efficiency, and reliability. Quantitative benchmarking confirmed significant performance improvements in processing speed and scalability, while qualitative testing validated the system's seamless workflow and ease of use.

The results collectively verify that all major functional requirements were achieved, including automatic exam compilation, duplication detection, and grammar checking within a secure local environment. These findings confirm the system's readiness for real-world deployment and provide the basis for further discussion in the next chapter.

Chapter 6: Conclusion

The development of **AUREX – Exam Paper Generation System** successfully addressed the long-standing inefficiencies in manual exam preparation at higher-education institutions. By integrating automation, natural-language processing, and customizable template generation into a single platform, the system achieved its core objective of reducing human workload while maintaining accuracy, fairness, and confidentiality in exam assembly.

The **proposed methodology**—comprising Excel-based question parsing, grammar error detection, multi-tier similarity analysis, and an interactive editing interface—proved effective in overcoming the major pain points identified in the problem statement. Evaluation results demonstrated that AUREX not only met but exceeded its functional goals:

- **Automation and Efficiency:** The system reduced exam-paper preparation time from over 4 hours manually to roughly 10 minutes, achieving a 95.5% improvement in efficiency.
- **Accuracy and Quality Assurance:** The hybrid duplication-detection pipeline (TF-IDF + semantic embeddings) consistently identified overlapping or paraphrased questions with high precision, while the grammar-checking module improved linguistic clarity.
- **Extended Functionality:** An additional *exam paper similarity checking* feature was introduced during development—beyond the original project proposal—to assist coordinators in detecting question overlap across different exam papers. This enhancement significantly strengthens academic integrity and further streamlines quality control.
- **Usability and Reliability:** End-to-end workflow testing confirmed smooth interaction across modules—from data ingestion to Word export—validating both the technical robustness and user-friendliness of the platform.
- **Data Privacy:** Running entirely in a local environment eliminates risks associated with cloud storage, ensuring full compliance with academic confidentiality requirements.

Taken together, these findings confirm that the **proposed system successfully solved the initial problem** of fragmented and time-consuming exam compilation. AUREX provides a scalable, secure, and practical tool that can be immediately applied in real university environments to standardize and streamline exam generation. Moreover, the integration of **additional similarity analysis across exam papers** not only enhances its analytical depth but also demonstrates the system's potential for continuous feature expansion and institutional adaptability.

Future Work

While AUREX has proven effective within its defined scope, several opportunities remain for further enhancement and academic exploration:

1. **Multi-User and Role-Based Access:** Introduce separate portals for lecturers, coordinators, and administrators to enable collaborative question contribution and review workflows.
2. **Centralized Question Repository:** Implement a secure local or on-premises database to allow long-term question management, version tracking, and reuse analytics while preserving privacy.
3. **Enhanced NLP Models:** Incorporate transformer-based large language models (e.g., BERT, GPT) to improve semantic understanding in duplicate detection, question classification, and difficulty prediction.
4. **Automated Question Quality Scoring:** Extend the system to assess question difficulty, cognitive level (Bloom's taxonomy), and content alignment to further support balanced exam construction.
5. **Cross-Platform and Cloud Integration:** Develop hybrid deployment options that combine local data control with optional institutional cloud synchronization for scalability across departments.
6. **AI-Assisted Question Generation:** Explore integration of AI-based question-generation tools that can propose new items derived from course materials, providing coordinators with richer question pools.

Final Remarks

In conclusion, **AUREX** has demonstrated that intelligent automation can meaningfully transform traditional academic workflows without compromising human oversight or data security. The system establishes a strong foundation for future innovations in academic assessment technology—bridging manual expertise with intelligent automation toward more efficient, equitable, and scalable examination management.

References

[1] Python Software Foundation. *Python 3.12 Documentation.*

From https://docs.python.org/3.12

[2] Node.js Foundation. *Node.js v22 Documentation.*

From https://nodejs.org/docs/latest/api

[3] React.js Team. *React 18 Documentation.*

From https://react.dev

[4] LanguageTool. *LanguageTool Open-Source Grammar and Style Checker.*

From https://languagetool.org

[5] scikit-learn Developers. *scikit-learn Machine Learning Library Documentation.*

From https://scikit-learn.org/stable

[6] Jinja2 Project. *Jinja2 Template Engine Documentation.*

From https://jinja.palletsprojects.com

[7] Flask Framework Team. *Flask Documentation.*

From https://flask.palletsprojects.com

[8] Pandas Development Team. *pandas: Data Analysis Library Documentation.*

From https://pandas.pydata.org