

# Aurex Exam Generator - Complete User Manual

**Version:** 1.0.0

**Built with:** Next.js, Python Flask

**License:** Academic Use

---

## Contents

Overview .....	1
Key Features.....	1
System Requirements .....	2
Installation.....	2
Quick Start Guide .....	4
User Interface Guide .....	5
Troubleshooting .....	10
FAQ.....	14
Version History and Changelog .....	15
License and Legal .....	16
Contact Information.....	17

---

## Overview

**Aurex Exam Generator** is a comprehensive desktop application designed for educators to create, manage, and generate professional exam papers. The system provides intelligent features including grammar checking, duplicate question detection, similarity analysis, and automated document generation.

## Key Features

- **Excel Question Bank Processing** - Upload and parse structured question banks
- **Advanced Grammar Checking** - Integrated LanguageTool grammar validation

- **Intelligent Duplicate Detection** - AI-powered similarity analysis
  - **Question Editor** - Interactive editing with real-time validation
  - **Multiple Question Types** - Support for MCQ, True/False, Matching, and Written questions (Short/Long)
  - **Document Generation** - Automated Word document creation for exams and answer keys
  - **Image Support** - Upload and embed images in questions
  - **Template Management** - Custom Word document templates
  - **Local Processing** - All data processed locally for security
- 

## System Requirements

### Prerequisites

- **Operating System:** Windows 10/11, macOS 10.15+, or Linux Ubuntu 18.04+
- **Python:** Version 3.8 or higher
- **Node.js:** Version 18 or higher (LTS recommended)
- **Memory:** Minimum 4GB RAM (8GB recommended)
- **Storage:** At least 2GB free space
- **Internet:** Required for initial setup and grammar checking service

### Browser Compatibility

- Google Chrome (recommended)
  - Microsoft Edge
- 

## Installation

**Git Hub Link** - <https://github.com/juliaizbroke/SeniorProject1>

### Windows Installation (Recommended)

## **Method 1: Automated Setup (Recommended)**

1. **Download and extract the project (ZIP file)** to your desired location
2. **Navigate to project folder:** "C:\path\to\SeniorProject1"
3. **Run (Double-click) the setup script:** setup-and-run.bat

The script will automatically:

- Check Python and Node.js installation
- Create Python virtual environment
- Install all backend dependencies
- Install frontend dependencies
- Start both servers
- Open the application in your browser

## **Method 2: Manual Installation**

If the automated script fails, follow these steps:

1. **Install Python 3.8+**
  - Download from [python.org](https://python.org)
  - **⚠️ Important:** Check "Add Python to PATH" during installation
2. **Install Node.js 18+**
  - Download LTS version from [nodejs.org](https://nodejs.org)
  - This includes npm package manager
3. **Open 2 command prompts (1 for backend & 1 for frontend)** in the project folder.
4. **Setup Backend:**
  - cd backend
  - python -m pip install --upgrade pip
  - pip install -r requirements.txt
5. **Setup Frontend:**

- cd frontend
  - npm install **OR** pnpm install
- 

## Quick Start Guide

### First-Time Users

#### 1. Run Setup:

- Windows: Double-click **setup-and-run.bat**

#### 2. Wait for Services:

- Backend starts on <http://localhost:5001>
- Frontend opens at <http://localhost:3000>
- First startup takes 15-20 seconds (LanguageTool initialization)

#### 3. Access Application:

- Browser should open automatically
- If not, manually navigate to <http://localhost:3000>

### Subsequent Use

For faster startup after initial setup: Double-click **quick-start.bat**

This skips dependency installation and starts servers directly.

### Basic Workflow

1. **Upload Question Bank** - Excel file with structured questions
  2. **Review Questions** - Grammar errors highlighted automatically
  3. **Handle Duplicates** - Resolve any detected duplicate questions
  4. **Edit Questions** - Make necessary corrections using the editor
  5. **Generate Documents** - Create exam papers and answer keys
-

# User Interface Guide

## Landing Page



- **Get Started Button** - Navigate to main application

## Home Page Features

This screenshot shows the main interface of the Aurex application. It is divided into several sections: 1. Top Left: 'Upload Excel File (Question Bank)' with a 'Download Excel Template' button and a 'Processing Options' section containing checkboxes for 'Enable Duplicate Detection' and 'Enable Grammar Checking'. 2. Top Right: 'Upload Word Template (Optional)' with a 'Download Word Template' button and a 'Word Template Requirements' section. 3. Center: A large area for file processing, showing a progress bar for an 'Excel File Ready' file named 'QuestionBank-Template-V1\_duplicateQ.xlsx' (0.02 MB). 4. Bottom Right: A modal window titled 'Processing Excel File' with a progress bar, a 'Did You Know?' section about question types, and navigation arrows. The overall layout is clean and organized, designed for ease of use.

## File Upload Section

- **Download Template** - Download Excel template
  - **In Case** - get manually from  
C:\..\SeniorProject1\backend\processing\templates\download
- **Drag & Drop Area** - Upload Excel question bank files

- **Progress Tracking** - Real-time upload progress
- **File Validation** - Automatic format checking

## Processing Options

- **Enable Duplicate Detection** - AI-powered similarity analysis
- **Enable Grammar Checking** - LanguageTool integration

## Word Template Management

- **Download Template** - Download Excel template
- **Default Templates** - Built-in exam paper templates
- **Custom Upload** - Upload your own Word document templates
- **Template Selection** - Choose active template for document generation

## Navigation Pages

The screenshot shows the 'Aurex Exam Generation System' interface. At the top, there's a header bar with the 'Aurex' logo and 'Exam Generation System'. Below it, the main interface has two main sections:

- Select Categories**: A section where users can choose the number of questions from each category to include in their exam. It includes a note about possible duplicate questions detected.
- Question Selection by Category**: A detailed table for selecting questions by category. The table has columns for Category, Available, and Select. For the 'Multiple Choice' category, the 'Available' count is 2 and the 'Select' count is 2. For 'True/False', 'Matching', and 'Written Questions', the 'Available' counts are 0, 0, and 0 respectively. There's also a 'Written Questions' section below the table.

At the bottom right of the 'Question Selection' section, there's a blue button labeled 'APPLY FILTERS'.

## Category Selection (/category)

- **Question Type Distribution** - View questions by category
- **Question Count** - Statistics for each question type

- **Navigation Controls** - Proceed to editing

## Question Editor (/edit)

The screenshot shows two separate instances of the 'Edit Questions' page from the 'Aurex Exam Generation System'. Both instances feature a prominent orange warning box at the top stating 'Duplicate Question Groups Detected'.

**Top Instance:**

- Question Type: Multiple Choice
- Category: Food
- Question: Which of the following is a fruit?
- Options: A. Pork, B. Beef, C. Basil, D. Apple (D is selected)
- Difficulty: Very High (90%)

**Bottom Instance:**

- Question Type: True/False
- Category: home
- Question: Do chairs typically have four legs?
- Answer: T

A small note at the bottom of the bottom instance says: 'Please resolve all duplicate question groups before generating documents.'

- **Grammar Error Highlighting** - Visual indicators
- **Duplicates Handling** – Keep one or ignore duplicate questions
- **Image Upload** - Add images to questions
- **Question Type Management** - Handle MCQ, T/F, Matching, Written questions (Short/Long)

## Preview (/preview)

The screenshot shows the 'Preview Exam' page from the 'Aurex Exam Generation System'.

**Exam Details:**

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

**Preview Content:**

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

Question Selection

<b>8</b> Multiple Choice	<b>1</b> TrueFalse	<b>2</b> Matching	<b>4</b> Written
-----------------------------	-----------------------	----------------------	---------------------

**Section A: Multiple Choice Questions (8 points)**

1. Which of the following is a fruit?  
 A. Pork  
 B. Beef  
 C. Basil  
 D. Apple  
 E. None of the Above  
Category: food
2. Which of these animals lives in water?  
 A. Camel  
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

**GENERATE PAPER** →

[BACK TO HOME](#)

[UPLOAD EXCEL AGAIN](#)

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

[DOWNLOAD EXAM PAPER](#)

[DOWNLOAD ANSWER KEY](#)

[BACK TO HOME](#) | [UPLOAD EXCEL AGAIN](#)

- **Document Preview** - View generated exam papers
- **Answer Key Preview** - Review answer key format
- **Download Options** - Direct download buttons
  - **In Case Field Values are not updated:** Open downloaded Word exam paper and press **CTRL + A + F9**

## Landing Page

The screenshot shows a landing page with a search icon at the top. Below it is a title 'Similarity Checking' with a subtitle: 'Maintain exam integrity with intelligent similarity detection. Compare multiple exam papers to identify duplicates, ensure consistency, and uphold academic standards across all assessments.' A 'How It Works' section follows, featuring three numbered steps: 'Upload Papers', 'Analyze Content', and 'Get Results'. Each step has a brief description. Below this is a 'Key Features' section with a bulleted list of six items. At the bottom is a large blue button labeled 'Start Similarity Check'.

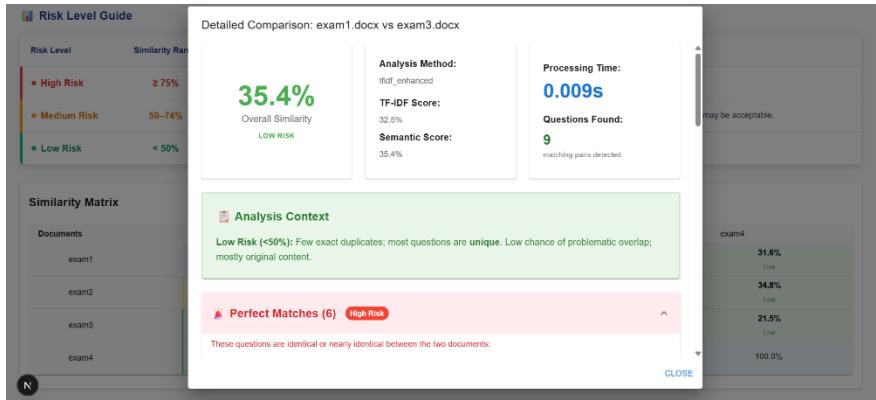
- **Start Similarity Check Button** - Navigate to similarity check page

## Similarity Check (/similarity)

The screenshot shows the 'Exam Papers Similarity Checker' page. It includes a 'Upload Exam Papers' section with a file upload area and a 'Drag and drop your Word documents here' instruction. Below this is a 'Risk Level Guide' table and a 'Similarity Matrix' table. The Risk Level Guide table defines three ranges: High Risk (2-75%), Medium Risk (50-74%), and Low Risk (< 50%). The Similarity Matrix table shows comparison results between four documents: exam1, exam2, exam3, and exam4. The matrix cells contain similarity percentages and risk levels (Low, Medium, High).

Risk Level	Similarity Range	Interpretation
High Risk	2-75%	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.

Documents	exam1	exam2	exam3	exam4
exam1	100.0% Low	41.5% Medium	35.4% Low	31.6% Low



- **Drag & Drop Area** - Upload multiple Word exam papers (old questions)
  - **Analysis Results** - Matrix of similarities with interactive cells
  - **Similarity Scores** - Numerical similarity ratings
- 

## Troubleshooting

### Installation Issues

#### Problem: "Python is not installed" error

##### Solution:

1. Download Python 3.8+ from [python.org](https://python.org)
2. **⚠ CRITICAL:** Check "Add Python to PATH" during installation
3. Restart Command Prompt/Terminal
4. Verify with: python --version

#### Problem: "Node.js is not installed" error

##### Solution:

1. Download Node.js LTS from [nodejs.org](https://nodejs.org)
2. Install with default settings
3. Restart Command Prompt/Terminal
4. Verify with: node --version and npm --version

#### Problem: “Dependencies” error

## **Solution:**

```
# Ensure pip is updated  
python -m pip install --upgrade pip  
  
# Install dependencies  
pip install -r backend/requirements.txt
```

## **Runtime Issues**

### **Problem: Application window is blank**

#### **Solutions:**

1. Wait 15-20 seconds for backend initialization
2. Refresh browser with Ctrl+R (Windows) or Cmd+R (macOS)
3. Check browser console for error messages
4. Verify both servers are running in terminal windows

### **Problem: "Port already in use" error**

#### **Solutions:**

1. Close any other instances of the application
2. Kill processes using ports 3000 and 5001:
  3. # Windows
  4. netstat -ano | findstr :3000
  5. netstat -ano | findstr :5001
  6. taskkill /PID <process\_id> /F
  - 7.
  8. # macOS/Linux
  9. lsof -ti:3000 | xargs kill -9
  10. lsof -ti:5001 | xargs kill -9
11. Restart your computer if issue persists

### **Problem: Grammar checking is slow or fails**

### **Solutions:**

1. **First startup:** LanguageTool downloads ~300MB data (one-time)
2. **Subsequent use:** Should be much faster
3. **Connection issues:** Ensure internet connectivity
4. **Firewall:** Allow Python applications through firewall
5. **Antivirus:** Whitelist the project directory

### **Problem: File upload fails**

### **Solutions:**

1. **File format:** Ensure Excel file (.xlsx) format
2. **File size:** Check file is under reasonable size limit
3. **Permissions:** Ensure read permissions on file
4. **Antivirus:** Temporarily disable real-time scanning
5. **Disk space:** Ensure sufficient free space in project directory

### **Problem: Generated documents are corrupted**

### **Solutions:**

1. **Template issues:** Try using default template
2. **Permissions:** Ensure write permissions in output directory
3. **Antivirus:** Whitelist output directory
4. **Dependencies:** Reinstall Python packages:
5. pip install --force-reinstall python-docx docxtpl

## **Performance Issues**

### **Problem: Slow duplicate detection**

### **Solutions:**

1. **Memory usage:** Close other applications
2. **Hardware:** Use machine with more RAM if available
3. **Optimization:** Process smaller batches of questions

## **Problem: High memory usage**

### **Solutions:**

1. **Cleanup:** Regularly use cleanup functions in app
2. **Restart:** Restart application periodically for large workflows
3. **Monitoring:** Use Task Manager/Activity Monitor to track usage

## **Network Issues**

### **Problem: Cannot access localhost:3000**

### **Solutions:**

1. **Wait longer:** Frontend takes time to compile and start
2. **Port conflict:** Try alternative port:
  - cd frontend
  - npm run dev -- --port 3001
3. **Firewall:** Configure firewall to allow local connections
4. **Browser cache:** Clear browser cache and cookies

### **Problem: Backend API calls fail**

### **Solutions:**

1. **Backend status:** Verify backend is running on port 5001
2. **CORS issues:** Check browser console for CORS errors
3. **Environment:** Verify .env.local has correct backend URL
4. **Network:** Disable VPN or proxy if active

## **Data Issues**

### **Problem: Excel file parsing fails**

### **Solutions:**

1. **Format validation:** Use provided Excel templates
2. **Sheet names:** Ensure "Info" and other Questions sheets exist
3. **Data integrity:** Check for empty cells in required columns

4. **Encoding:** Save Excel file with UTF-8 encoding
5. **Version:** Use Excel 2016+ or LibreOffice Calc

#### **Problem: Generated documents missing content**

#### **Solutions:**

1. **Template variables:** Verify template contains required placeholders
  2. **Question data:** Check question parsing completed successfully
  3. **Image paths:** Ensure image files are accessible
  4. **Metadata:** Verify Info sheet contains all required fields
- 

## FAQ

### **General Questions**

**Q: Is my data secure?** A: Yes! All processing happens locally on your computer. No data is sent to external servers except for grammar checking via LanguageTool, which only processes text snippets for validation.

**Q: Can I use this offline?** A: Partially. The application works offline except for grammar checking, which requires internet connectivity for LanguageTool service.

**Q: What file formats are supported?** A: **Input:** Excel (.xlsx), Word templates (.docx), Images (JPEG, PNG) **Output:** Word documents (.docx), HTML previews

**Q: How many questions can I process at once?** A: The system can handle hundreds of questions, but performance depends on your hardware. For best results, process 50-200 questions at a time.

**Q: Can I customize the exam paper format?** A: Yes! Upload your own Word document templates with custom formatting, headers, footers, and styling. However, we kindly advise that if you are not familiar with Jinja codes, please avoid removing or editing them to ensure everything works smoothly.

### **Technical Questions**

**Q: Why is the first upload slow?** A: LanguageTool downloads language models (~300MB) on first use. Subsequent uses are much faster as models are cached locally.

**Q: Can I run this on a server?** A: While technically possible, the application is designed for desktop use. Server deployment would require additional configuration and security considerations.

**Q: How accurate is the duplicate detection?** A: The similarity detection is highly accurate, using semantic analysis rather than simple text matching.

**Q: Can I batch process multiple Excel files?** A: Currently, the system processes one Excel file at a time. You can combine multiple question banks into a single Excel file for processing.

## Usage Questions

**Q: How do I create the Excel question bank?** A: Use the provided Excel templates or follow the structure documented in this manual. Include "Info" and other Question Types sheets with proper column headers.

**Q: Can I add images to questions?** A: Yes! Use the image upload feature in the question editor to add visual elements to any question type.

**Q: How do I handle grammar suggestions?** A: Grammar errors are highlighted in yellow in editing page. Use the question editor to review and apply suggestions manually.

## Troubleshooting Questions

**Q: The application won't start. What do I do?** A: 1) Verify Python and Node.js are installed, 2) Run as Administrator on Windows, 3) Check antivirus isn't blocking, 4) Try manual installation steps.

**Q: Generated Word documents won't open.** A: This usually indicates a template or formatting issue. Try using the default template first, then check your custom template for errors.

**Q: Images aren't showing in generated documents.** A: Ensure images are uploaded through the application and paths are correctly referenced. Check that image files haven't been moved or deleted.

---

## Version History and Changelog

### Version 1.0.0 (Current)

- **Initial Release** - Complete exam generation system
- **Core Features** - Excel parsing, grammar checking, duplicate detection

- **Document Generation** - Word paper and answer key creation
- **Template System** - Custom Word document templates
- **Image Support** - Question image upload and embedding
- **Web Interface** - Responsive Next.js frontend
- **Local Processing** - Privacy-focused local data handling

## Planned Enhancements

- **Performance Optimization** - Faster processing for large question banks
- 

# License and Legal

## Academic Use License

This software is provided for educational and academic purposes. Commercial use requires separate licensing agreement.

## Third-Party Dependencies

- **LanguageTool** - Grammar checking service
- **Flask** - Web framework (BSD License)
- **React/Next.js** - Frontend framework (MIT License)
- **Material-UI** - Component library (MIT License)
- **pandas** - Data processing (BSD License)
- **python-docx** - Document generation (MIT License)

## Data Privacy

- All question data are processed locally
- Grammar checking sends only text snippets to LanguageTool
- No personal data transmitted to external servers
- Temporary files automatically cleaned up

## Security Notice

- Run only on trusted networks
  - Scan uploaded files for malware
  - Keep dependencies updated for security patches
  - Use antivirus software for additional protection
- 

## Contact Information

### Development Team – The Three Must-code-tears

- **Project Name:** Senior Project 1 (CSX3010-SP1)
- **Institution:** AU (Academic University)
- **Academic Year:** 3rd Year, 2nd Semester
- **Project Advisor:** Asst. Prof. Dr. Rachsuda Setthawong
- **Team Members**
  - 6530207 – Zwe La Pyae (CS)
  - 6530213 – Thet Myat Noe Thwin (CS)
  - 6530203 – Khant Min Lwin (IT)

### Technical Support

For technical assistance with Aurex Exam Generator:

1. Check this user manual thoroughly
  2. Review troubleshooting section
  3. Contact development team with detailed error information
  4. Include system specifications and steps to reproduce issues
- 

### Thank you for using Aurex Exam Generator!

*This manual is comprehensive and covers all aspects of the system. For the most up-to-date information, always refer to the latest version of this documentation.*

---

*Document Version: 1.0.0 | Last Updated: September 2025 | Total Pages: Comprehensive User Guide*