Instructions

WASPL - Digital Assessment Platform (Alpha)
Status: Alpha
License: AGPL v3
Technology Stack: Node.js, Vue 3, MongoDB, Docker

? Table of Contents

? Overview

WASPL is an open-source digital assessment platform designed to create, manage, and deliver interactive online tests. Built with modern technologies (Node.js, Vue 3, MongoDB, Docker), it provides educators with powerful tools to design rich and flexible testing experiences.

⚠ **Pre-Alpha Version**: This software is in active development and not yet ready for production use.

? Features

- **Waspleditor**: Create tests using 12 interactive question types.
- **WasplTestRunner**: Deliver assessments in EXAM or LEARNING mode.
- Rich interactions: drag-and-drop, clickable images, gap-fills, and more.
- **AI Integration**: Automatic scoring and test generation.
- **Internationalization**: Supports 6 languages (EN, FR, ES, DE, IT, PT).
- **MongoDB**: Stores responses and metadata.
- **Strict ESM**: Modern modular imports everywhere.

? Architecture

WASPL consists of two main applications:

- `waspleditor`: Admin/editor interface to build tests.
- `waspltestrunner`: Interface for test takers.

Both apps use a shared backend and a MongoDB database. Services are orchestrated with Docker.

Full documentation is available at: https://waspl-wiki.wiquid.fr/

? Getting Started

? Local Development (no Docker)

1. Clone the repository:
   bash
   ```
   git clone https://github.com/janfix/waspl.git
   cd waspl
   ```
2. Install dependencies:
   bash
   ```
   cd waspleditor && npm install
   cd ../waspltestrunner && npm install
   ```
3. Start MongoDB manually (if needed), then:
   bash
   ```
   cd ../waspleditor && npm run dev
   cd ../waspltestrunner && npm run dev
   ```

Pre-configured `.env` files are included with safe default values.

? Docker Usage

? Development Mode (hot-reload)

Command:

`make dev`

- Waspleditor: http://localhost:5173
- WasplTestRunner: http://localhost:5174

Services wait for MongoDB before launching.

? Production Mode (with NGINX)

Command:

`make prod`

- Single entry point: http://localhost/
  - `/editor/` for the editor
  - `/runner/` for test-taking

NGINX handles redirection, security headers, and gzip compression.

?? Available `make` Commands

Command	Description
`make dev`	Start all services in dev mode
`make prod`	Start all services in production mode
`make down`	Stop and remove containers
`make reset-db`	Remove MongoDB local data
`make reset-all`	Remove all containers, data and Docker images

?? Environment Variables

? Quick Setup (Development)

Everything works out of the box after cloning:

`npm run dev`

? Configuration File Structure

File	Included	Description

`.env`	✅ Yes	Base development configuration
`.env.docker`	✅ Yes	Docker-specific configuration
`.env.template`	✅ Yes	Example with placeholders
`.env.local`	❌ No	Local custom variables (ignored)
`.env.production`	❌ No	Sensitive production configuration

? Production Configuration

Create a `.env.local` file with real values:

```
VITE_CHATGPT_API=sk-your-openai-key
VITE_SYSTEM_TOKEN_FOR_TESTRUNNER=your-jwt-token
JWT_SECRET=your-strong-secret
MONGO_URI=mongodb://your-server:27017/wasplData
```

Or use system environment variables:

```
export VITE_CHATGPT_API="sk-your-key"
export JWT_SECRET="your-secret"
```

?? Security

- ✅ Development files use safe defaults and are included
- ❌ Real secrets are excluded from Git
- ✅ Templates are available to simplify setup

? Interaction Types

WASPL supports 12 interaction types:

- **Choice**: Classic single or multiple choice
- **Choice Block**: Choice with rich media support
- **Text Puzzle**: Drag-and-drop gap-filling puzzle
- **Text Gaps**: Traditional fill-in-the-blank text
- **Short Answer**: Free-form answer with AI scoring
- **Hot Spot**: Clickable image-based interactions
- **Order**: Sequencing with drag-and-drop
- **Match**: Match pairs between two columns
- **Making Pairs**: Build and assign item pairs
- **Message**: Instructional text without input
- **No Editor**: Raw content mode
- **Default**: Custom base interaction

? Default User

Default credentials:

- Username: `admin`
- Password: `password`

? Project Structure

```
waspleditor/         → Vue3 + Vite app for test creation
  ├─ Dockerfile
  └─ Dockerfile.dev
waspltestrunner/     → Vue3 + Vite app for test execution
shared/              → Shared models and utils
media/               → Static files (images, audio)
scripts/             → Utility scripts
certbot/             → SSL certificate automation
docker-compose.yml   → Docker orchestration config
nginx.conf*          → NGINX config files
Makefile             → CLI helper commands
```

? Contributing

We welcome contributions!
Open an issue or submit a pull request.
Please follow the existing code style and use meaningful commit messages.

? Setup for Contributors

1. Clone the repo:
   ```
   git clone https://github.com/janfix/waspl.git
   cd waspl
   npm run dev
   ```
2. (Optional) Enable AI features:
   ```
   cp waspleditor/.env.template waspleditor/.env.local
   # Then add your OpenAI key
   ```
3. Recommended: use Docker
   ```
   make dev
   ```

? License

This project is licensed under the **AGPL v3**.
See the LICENSE file for details.

Revision #4
Created 14 May 2025 09:16:49 by Janfix
Updated 8 June 2025 10:06:47 by Janfix