Pathogenius

Real-Time Pathogen Detection & Clinical Analysis

User Guide

Version 1.0

This manual covers all features of the Pathogenius desktop application — from first login to advanced GPU-accelerated metagenomic analysis, AI-powered clinical summaries, and encrypted cloud synchronisation.

Contents

List of Figures

Chapter 1

Introduction

1.1 About Pathogenius

Pathogenius is a desktop application for **real-time pathogen detection** from metagenomic FASTQ sequencing data. It combines a modern Electron-based graphical interface with a powerful Snakemake/CLARK-l bioinformatics backend, supporting both local Docker-based CPU classification and GPU-accelerated edge computing on a Jetson Nano device.

Key capabilities at a glance:

- **Dual classification engines** — CLARK-l (CPU via Docker) or CU-CLARK-L (GPU via Jetson Nano)
- **Interactive visualisations** — Treemap, Sunburst, Sankey, and Radar charts
- **AI-powered clinical summaries** — local MedGemma 4B model runs fully offline
- **Encrypted cloud sync** — AES-256-GCM encrypted upload/download via Firebase
- **User management** — Firebase authentication with admin controls
- **Custom reference databases** — build and manage CLARK databases from custom genomes
- **Real-time system dashboard** — CPU, RAM, disk, and GPU monitoring

Important

Pathogenius is a **decision-support tool only**. Results are computational predictions based on sequence similarity. They must never be used as the sole basis for clinical diagnosis. Always confirm findings with certified clinical laboratory tests and consult a qualified healthcare professional.

1.2 Intended Audience

This user guide is written for:

- Researchers and bioinformaticians running metagenomic sequencing workflows
- Clinical laboratory staff using Pathogenius as a supplementary decision-support tool

- System administrators managing user accounts and database builds
- Field workers using the offline/guest mode and GPU edge-computing features

1.3 Document Conventions

Throughout this guide the following conventions are used:

Convention	Meaning
[Button Name]	Clickable button in the UI
Field Name	Input field or form element
`UI element`	Specific UI component or label
Active	Status badge

1.4 System Requirements

Component	Requirement	Notes
Operating System	Windows 10/11 (64-bit)	macOS/Linux supported with adjustments
Node.js	Version 22 or later	Required for the Electron frontend
Python	Version 3.8 or later	Required for Snakemake pipeline
Snakemake	Version 7 or later	`pip install snakemake`
Docker Desktop	Latest stable	Required for CPU classification
RAM	8 GB minimum	16 GB recommended for large files
Disk Space	20 GB minimum	For databases and result storage
Internet	Required for setup	Offline operation after initial setup
GPU mode (optional)		
Jetson Nano	With CU-CLARK-L	Connected via Tailscale VPN
Tailscale	Latest client	For secure SSH tunnel to Jetson Nano

Chapter 2

Getting Started

2.1 Installation Overview

2.1.1 Step 1 — Install Prerequisites

Before launching Pathogenius, ensure the following are installed on your system:

1. **Node.js 22+** — download from `https://nodejs.org/`
2. **Python 3.8+** — download from `https://www.python.org/`
3. **Snakemake** — install via pip:

   ```
   pip install snakemake pyyaml
   ```

4. **Docker Desktop** — download from `https://www.docker.com/`. Ensure Docker is running before starting any CPU-mode analysis.

2.1.2 Step 2 — Install Frontend Dependencies

Open a terminal, navigate to the `frontend/` directory, and run:

```
cd frontend
npm install
```

This installs Electron, Firebase SDK, `node-llama-cpp`, `keytar`, `systeminformation`, and all other dependencies.

2.1.3 Step 3 — Build the Reference Database (First-Time)

Place pathogen genome FASTA files (`.fna`, `.fasta`, `.fa`, `.fsa`, or gzipped variants) into the `Patho-genius/clark_db/` directory. Then run:

```
cd Patho-genius
python build_clark_db.py
```

This will automatically:

1. Scan all genome sources listed in `config.yaml`
2. Resolve taxonomy IDs via NCBI lookup or reads-mapping

3. Copy genomes into `clark_db/Custom/`
4. Download NCBI taxdump (~55 MB)
5. Run `set_targets.sh` inside Docker to produce `targets.txt`

> **Note**
>
> The database build requires Docker to be running and an active internet connection for NCBI taxonomy lookups. This step only needs to be performed once, unless you add new genome sequences.

2.1.4 Step 4 — Launch the Application

```
cd frontend
npm start
```

The Pathogenius window will open automatically.

2.2 First Launch

On first launch, the **Login** screen is displayed. You can either:

- **Create a new account** to access all features including cloud synchronisation.
- **Log in with existing credentials** if your administrator has already created an account for you.
- **Use Guest Mode** for immediate offline access without registration.

Chapter 3

Authentication

3.1 Login

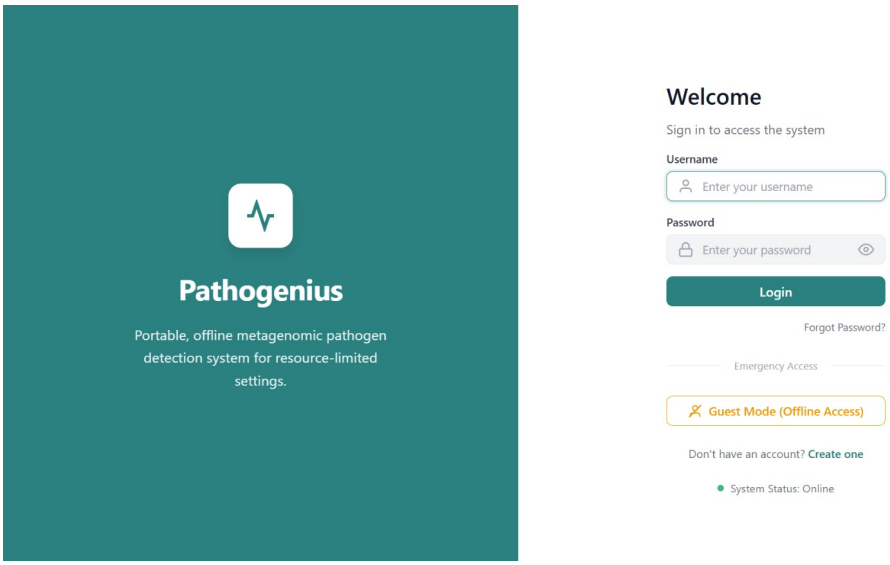

Figure 3.1: The Pathogenius login screen with username/password fields and guest mode option.

The login screen features a two-panel layout: a branded information panel on the left and the login form on the right.

3.1.1 Logging In

1. Enter your *Username* or email address.
2. Enter your *Password*. Use the 👁 (eye) icon to toggle password visibility.
3. Click [Login].

On successful login you are taken directly to the **Dashboard**.

3.1.2 Common Login Errors

Error	Resolution
Invalid credentials	Check username and password. Passwords are case-sensitive.
Email not verified	Check your inbox for the verification email. Click **[Resend Verification Email]** if needed.
Account suspended	Contact your system administrator.

3.1.3 Forgot Password

1. Click the **[Forgot Password?]** link below the login form.
2. Enter your registered email address or username in the dialog.
3. Click **[Send Recovery Link]**.
4. Check your inbox for the password reset email and follow the link.

3.2 Creating a New Account

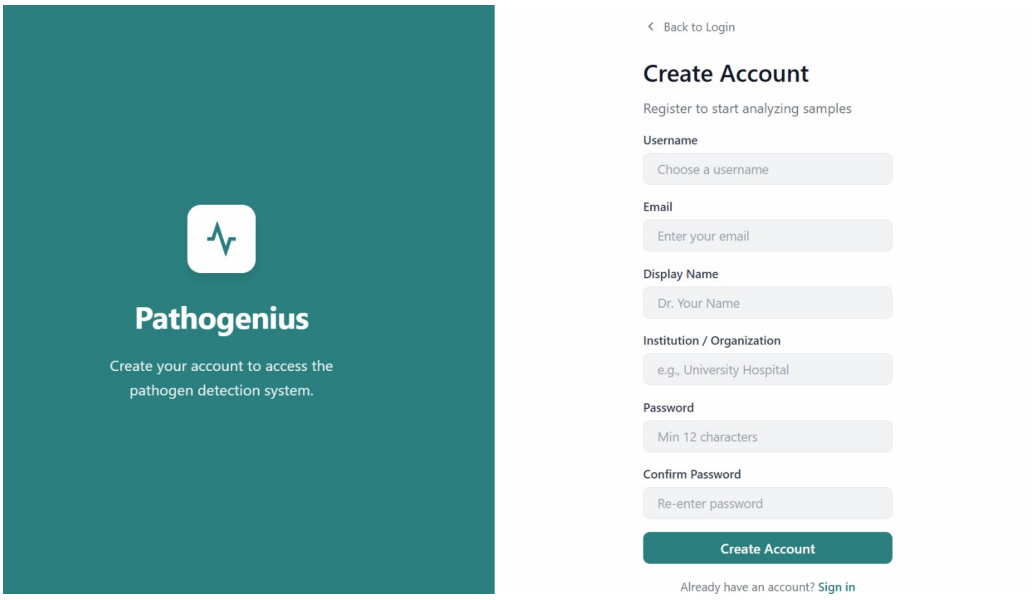

Figure 3.2: Account registration form with password strength indicator.

1. On the Login screen, click **[Create Account]**.
2. Fill in the registration form:
   - *Username* — at least 3 characters
   - *Email* — a valid email address (used for verification and password reset)
   - *Display Name* — optional; defaults to your username
   - *Institution/Organisation* — optional

- *Password* — see requirements below
- *Confirm Password* — must match the password field exactly

3. Click **[Create Account]**.
4. A verification email is sent to your address. Click the link in the email.
5. Return to Pathogenius and click **[I've Verified My Email]**.
6. You are redirected to the login screen with a success message.

Password Requirements

Your password must meet **all** of the following criteria:

✔ At least 12 characters in length

✔ At least one uppercase letter (A–Z)

✔ At least one number (0–9)

✔ At least one special character (`!@#$%^&*()_+-=[]...`)

A real-time strength indicator and checklist are shown as you type.

3.3 Guest Mode (Offline Access)

Guest mode grants immediate access without an account. Click **[Guest Mode (Offline Access)]** on the login screen.

Warning

Guest mode has the following limitations:
- Results are saved locally only — no cloud synchronisation
- Cloud upload and download are disabled
- Password change and admin features are unavailable

To unlock all features, create a registered account.

Chapter 4

Application Layout

4.1 Main Window Structure

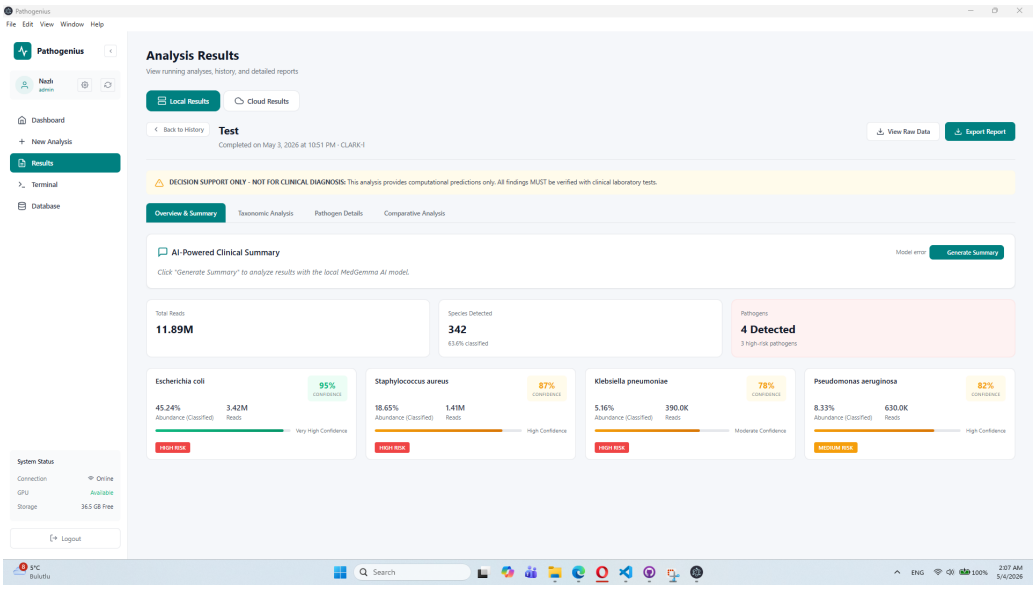

Figure 4.1: Main window showing the sidebar navigation and content area.

The Pathogenius window is divided into two main areas:

- **Sidebar** (left) — persistent navigation panel present on all screens.
- **Content area** (right) — displays the active page.

4.2 Sidebar Navigation

4.2.1 Brand & User Card

At the top of the sidebar:

- The **Pathogenius** logo and application name.
- A **User Card** showing your display name and role (e.g., *Researcher*).
- A ⚙ (Settings) icon button to jump to the Settings page.

- A (Refresh) icon button to reload system status information.

4.2.2 Navigation Menu

Icon	Page	Purpose
	Dashboard	System overview, running analyses, recent history
	New Analysis	Wizard to launch a new classification run
	Results	View and manage all analysis results
	Terminal	Embedded terminal for advanced pipeline control
	Database	Build and manage reference databases

The active page is highlighted. A numeric badge on *Results* shows the count of currently running analyses.

4.2.3 System Status Widget

At the bottom of the sidebar a compact status widget shows:

- **Connection** — Online / Offline indicator.
- **GPU** — Available / Not detected.
- **Storage** — Free disk space in GB.

In Guest Mode a yellow notice banner reads *"Results saved locally (no cloud sync)"*.

The **[Logout]** button is at the very bottom of the sidebar.

Chapter 5

Dashboard

5.1 Overview

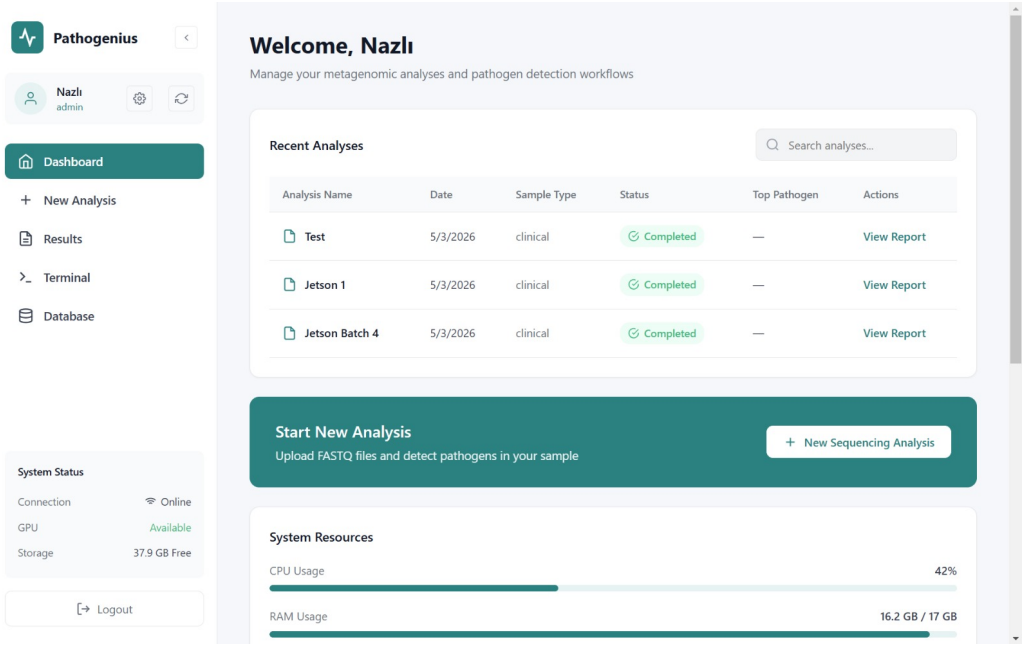

Figure 5.1: Dashboard showing system resources, analysis summary, and recent analyses table.

The Dashboard is the home screen, providing a real-time snapshot of system health and analysis activity.

5.2 Running Analysis Banner

When an analysis is actively processing, a prominent banner appears at the top of the Dashboard showing:

- The **analysis name**
- The current **status message** (e.g., *"Classifying reads...")*
- An animated **progress bar** (0–100%)

- A [View Details] button that navigates directly to the Results page

5.3 System Resources Card

Displays real-time hardware metrics:

- **CPU Usage** — percentage utilisation with a colour-coded progress bar
- **RAM Usage** — gigabytes used out of total, with a progress bar
- **GPU Status** — GPU model name if detected, or *"No GPU detected"*

5.4 Storage Card

Shows available disk space with a progress bar indicating used vs. total capacity.

5.5 Analysis Summary Card

Four counters provide a quick health check:

Counter	Description
Total Analyses	All analyses ever run on this installation
Completed	Successfully finished analyses
Failed	Analyses that terminated with an error (shown in red)
Running	Analyses currently processing

5.6 Recent Analyses Table

Lists the 10 most recent analyses with columns for Name, Date, Sample Type, Status, Top Pathogen detected, and an action button. Use the *Search* box to filter by name. Click [View Report] to open the full detail view for any completed analysis.

5.7 Quick Actions

The [Start New Analysis] card at the bottom of the Dashboard navigates directly to the New Analysis wizard.

Chapter 6

Running a New Analysis

6.1 Overview

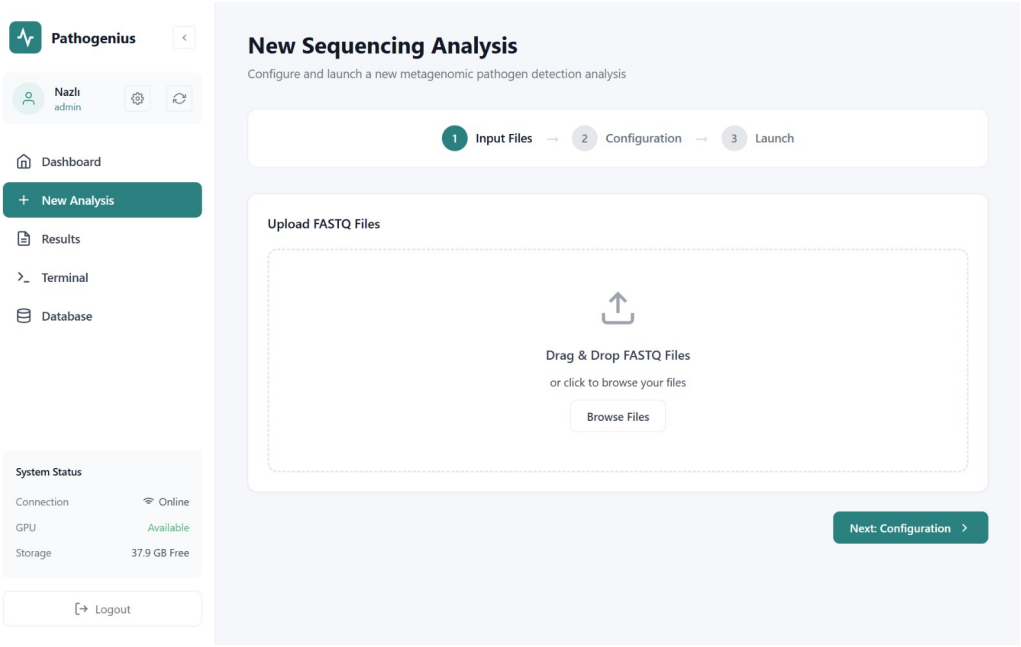

Figure 6.1: File selection step of the three-step analysis wizard.

The New Analysis wizard is a three-step process:

1. **Input Files** — select the FASTQ sequencing data
2. **Configuration** — choose the classification engine and analysis settings
3. **Review & Launch** — confirm all settings and start the run

A step indicator at the top of the page shows your current position. Click [Back] at any step to revise previous selections.

6.2 Step 1: Input Files

6.2.1 Selecting FASTQ Files

Two methods are available:

- **Drag and Drop** — drag one or more `.fastq` or `.fastq.gz` files directly onto the upload zone.
- **Browse** — click inside the upload zone to open a native file-picker dialog.

After selection the file list appears, showing each filename with a [Clear] button to reset the selection.

> **Note**
>
> Pathogenius accepts standard FASTQ format. Gzip-compressed files are supported. Only one file is required for a single-sample analysis.

Click [Next: Configuration] to proceed. This button is disabled until at least one file is selected.

6.3 Step 2: Configuration

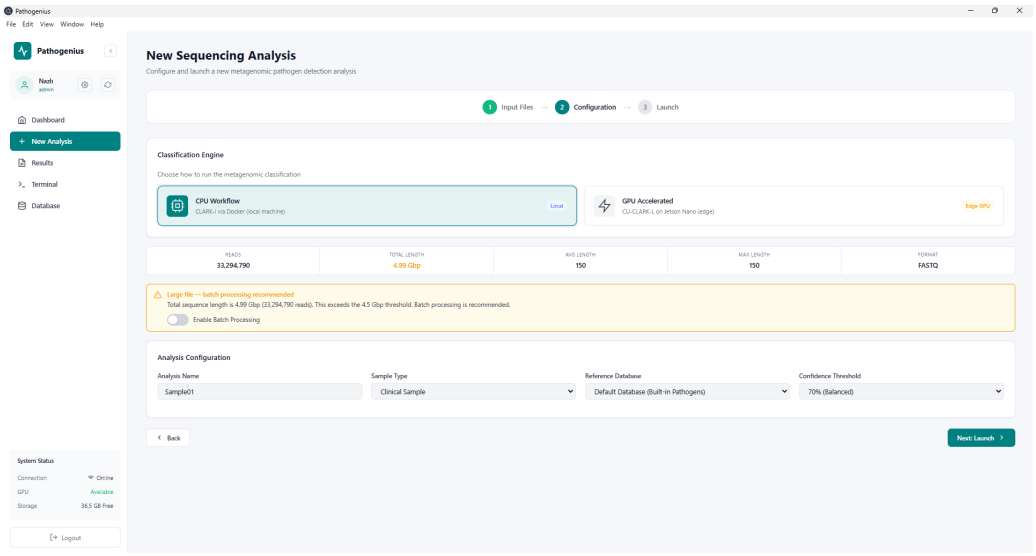

Figure 6.2: Configuration step showing engine selection cards, analysis settings form, and FASTQ statistics banner.

6.3.1 Engine Selection

Choose between two classification engines using the clickable cards:

Engine	Description	Requirement
CPU Workflow (default)	CLARK-l running inside a Docker container on the local machine. Suitable for most analyses.	Docker Desktop running
GPU Accelerated	CU-CLARK-L on the Jetson Nano edge device via Tailscale SSH. Significantly faster for large datasets.	Tailscale connected; Jetson Nano reachable

6.3.2 FASTQ Statistics Banner

After file selection, Pathogenius automatically analyses the first FASTQ file using SeqKit and displays:

- Total read count
- Total sequence length
- Average and maximum read length
- File format confirmation

6.3.3 Large File Warning and Batch Processing

If the total sequence length exceeds **4.5 Gbp**, a warning banner appears. Enable the *Batch Processing* toggle to split the file into two batches, which prevents memory overruns and reduces the risk of classification crashes.

Tip

For very large clinical FASTQ files we recommend enabling batch processing. It slightly increases total runtime but greatly improves reliability.

6.3.4 Analysis Configuration Form

Field	Description	Required
Analysis Name	A unique, descriptive name for this run (e.g., *Patient_001_Sample*). Used to identify the result later.	Yes
Sample Type	Category for the sample: Clinical Sample, Environmental, Food Safety, or Other. Affects report labelling.	Yes
Reference Database	The CLARK database to use for classification. Defaults to the built-in database. Custom databases appear here once built.	Yes
Confidence Threshold	Minimum confidence score to report a pathogen. Lower values increase sensitivity; higher values increase precision.	Yes

Warning

If you select a custom database that has not yet been built or activated, a warning is shown. The database will be rebuilt automatically before the analysis starts, which may add several minutes.

Click **[Next: Launch]** to proceed. The Analysis Name field must be filled before continuing.

6.4 Step 3: Review & Launch

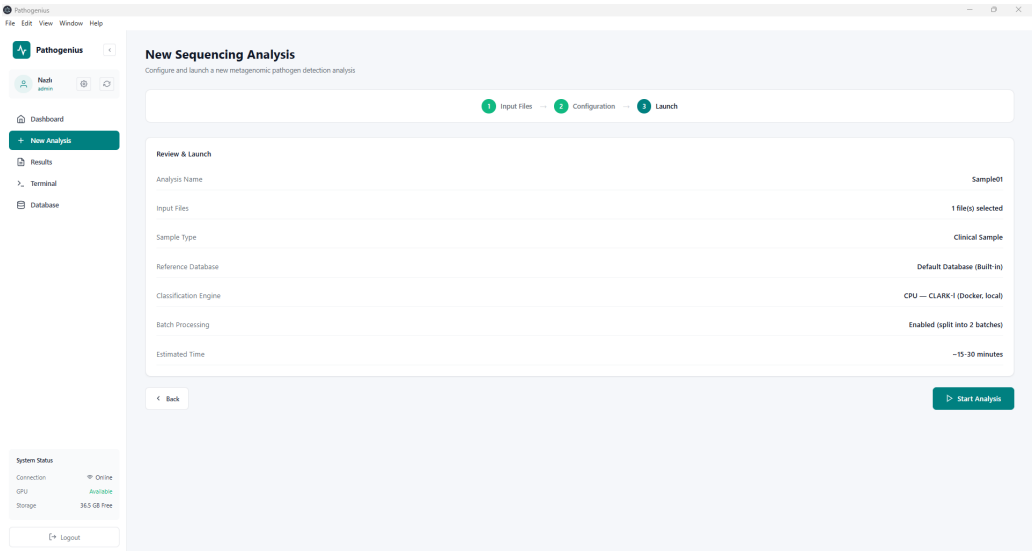

Figure 6.3: Review screen showing all selected settings before launching.

A read-only summary card displays every configuration choice:

- Analysis Name

- Input Files (count)
- Sample Type
- Reference Database
- Classification Engine (with icon)
- Batch Processing (Enabled / Disabled)
- Estimated Time (∼15–30 minutes for typical datasets)

Review the summary. If anything needs changing, click **[Back]**. When satisfied, click **[Start Analysis]**. The application navigates to the Results page and begins processing immediately.

6.5 Monitoring Progress

Once an analysis starts:

- A **progress banner** appears on the Dashboard with real-time status messages and a percentage bar.
- The **Running Analyses** section on the Results page shows the analysis card with pause, resume, and cancel controls.
- The sidebar *Results* badge increments to show the count of active runs.

6.5.1 Pausing and Resuming

Click ⏸ **[Pause]** on the running analysis card to pause the Snakemake pipeline. Click ▶ **[Resume]** to continue. Pausing is useful if system resources are needed for other tasks.

6.5.2 Cancelling an Analysis

Click the ✖ **[Cancel]** button on the running analysis card. A confirmation prompt appears. Cancellation stops the pipeline and marks the result as **Cancelled**. Partial outputs are discarded.

Chapter 7

Viewing Results

7.1 Results Page Overview

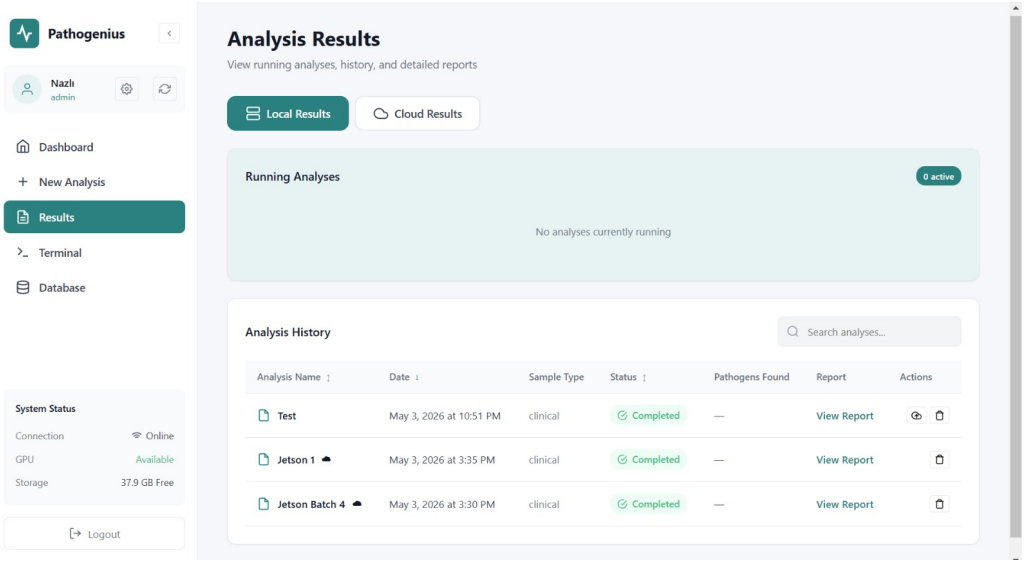

Figure 7.1: Analysis history table showing completed, failed, and running analyses.

The Results page is divided into two source tabs:

- **Local Results** — analyses stored on this machine.
- **Cloud Results** — analyses synchronised to Firebase Storage (registered users only).

7.2 Analysis History Table

The history table lists all local analyses with the following columns:

Column	Description
Analysis Name	Clickable name; opens the detail view
Date	Completion or start timestamp
Sample Type	Category selected at launch
Status	**Completed**, **Failed**, Running, or **Cancelled**
Pathogens Found	Count of detected pathogens, or "—" if none
Report	**[View Report]** button (completed analyses only)
Actions	Cloud upload and delete buttons

Use the *Search* box to filter by analysis name. A cloud icon (☁) appears next to analyses that have been uploaded.

7.2.1 Action Buttons

- **[Upload]** — encrypts and uploads the result to Firebase (registered users only).
- **[Delete]** — permanently removes the result from local storage. A confirmation dialog appears first.

7.3 Detailed Results View

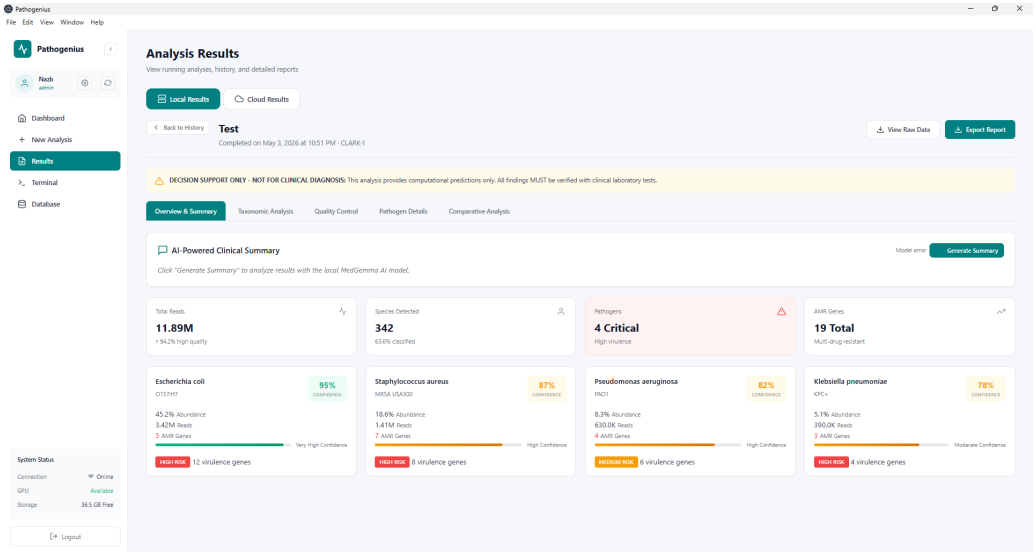

Figure 7.2: Pathogen cards, AI summary section, and chart containers for a completed analysis.

Clicking **[View Report]** or the analysis name opens the full detail view.

7.3.1 Header

- **[Back to History]** — returns to the analysis list.
- Analysis name as a large heading.
- Subtitle: *"Completed on [date] · [Classifier]"*.

- **[Export Report]** — generates a downloadable PDF report.
- **[View Raw Data]** — displays the raw JSON classification output.

Important

A red banner at the top of the detail view reads:
"DECISION SUPPORT ONLY — NOT FOR CLINICAL DIAGNOSIS"
Results are computational predictions and must be confirmed by a certified clinical laboratory.

7.3.2 Overview Tab — Pathogen Cards

Up to four pathogen cards are shown. Each card contains:

- Pathogen name
- Confidence badge with percentage
- **Metrics:** Abundance among classified reads (%), Read count
- Colour-coded confidence bar (green / amber / red)
- Risk level badge: **High Risk**, **Medium Risk**, or **Low Risk**
- Cards for high-risk pathogens have a red background tint

7.3.3 AI Summary Card

If AI summaries are enabled in Settings:

1. Click **[Generate Summary]**.
2. The local MedGemma 4B model analyses the results and produces a clinical interpretation.
3. The summary appears within 30–120 seconds depending on hardware.
4. Click **[Regenerate]** to produce an alternative summary.

Note

The MedGemma model runs entirely on your local machine. No data is sent to external servers. A GGUF model file must be present in `frontend/models/` for this feature to work.

7.3.4 Pathogen Details Tab

An expanded card for each detected pathogen with full metrics:

- Abundance among classified reads (%)
- Abundance among total reads (%)
- Read Count

- Confidence Score (%)
- Virulence Factors alert box with gene count
- NCBI Taxonomy ID

7.3.5 Charts Tab

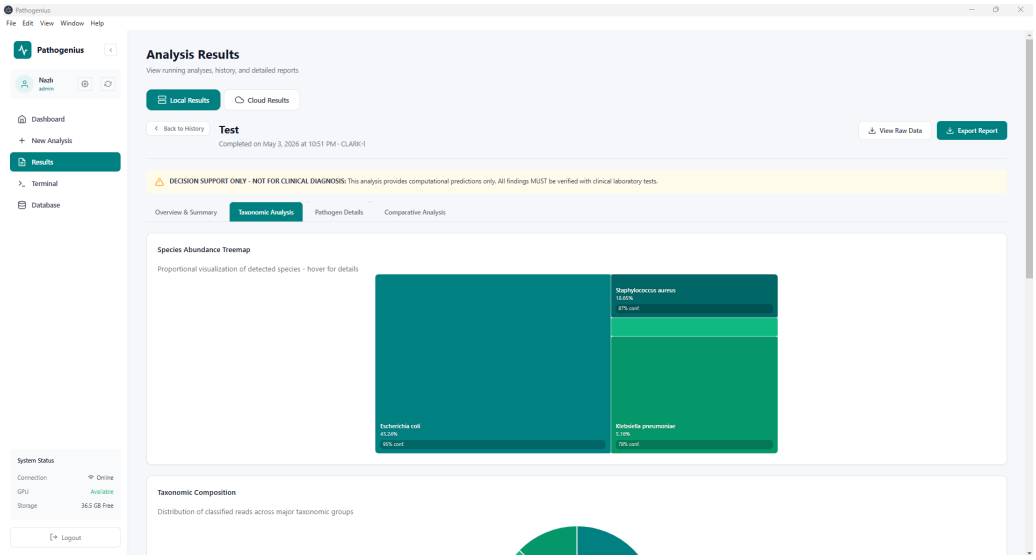

Figure 7.3: Charts Tab — Treemap, Sunburst, Sankey, and Radar charts.

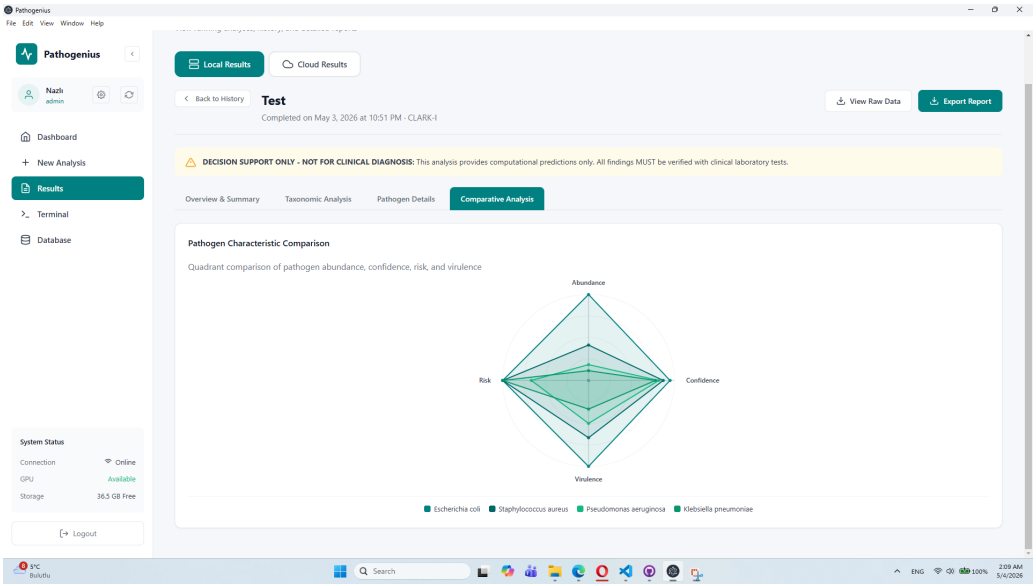

Figure 7.4: Charts Tab — comparative view across top pathogens.

Four interactive SVG charts visualise the classification results:

Chart	Type	Description
Treemap	Area chart	Proportional rectangles sized by pathogen abundance. Colour-coded by risk level. Hover for tooltips.
Sunburst	Ring chart	Hierarchical taxonomy view: Domain → Phylum → Class → Species. Click to drill down.
Sankey	Flow chart	Read flow: Total Reads → Classified/Unclassified → Taxonomic groups → Species.
Radar	Spider chart	Comparative view of top pathogens across five axes: Abundance, Confidence, Read Count, Virulence, AMR Genes.

All charts are generated from the classification JSON output and update automatically when confidence filters are changed in Settings.

7.4 Cloud Results

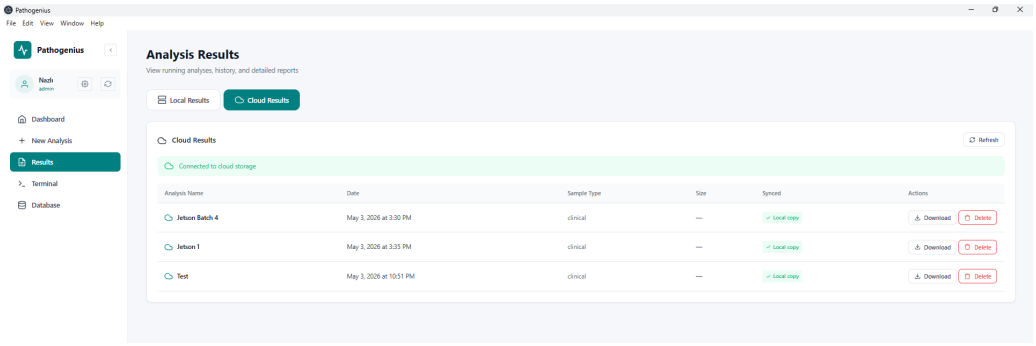

Figure 7.5: Cloud results list with download actions for registered users.

Switch to the **Cloud Results** tab to view analyses synchronised to Firebase Storage.

7.4.1 Cloud Status Banner

Shows whether the cloud connection is active: *"Connected to cloud storage"* or *"Unable to connect"*.

7.4.2 Cloud Results Table

Column	Description
Analysis Name	Cloud-stored analysis identifier
Date	Upload timestamp
Sample Type	Category
Size	Encrypted file size
Status	**Local copy** (downloaded) or **Cloud only**
Actions	[Download] and [Delete] buttons

Click [Download] to decrypt and store a cloud result locally. Click [Delete] to permanently remove a result from the cloud (local copies are not affected). Click [Refresh] to update the list from Firebase.

Note

Cloud features are unavailable in Guest Mode. A locked notice with login and register links is displayed instead.

Chapter 8

Database Management

8.1 Overview

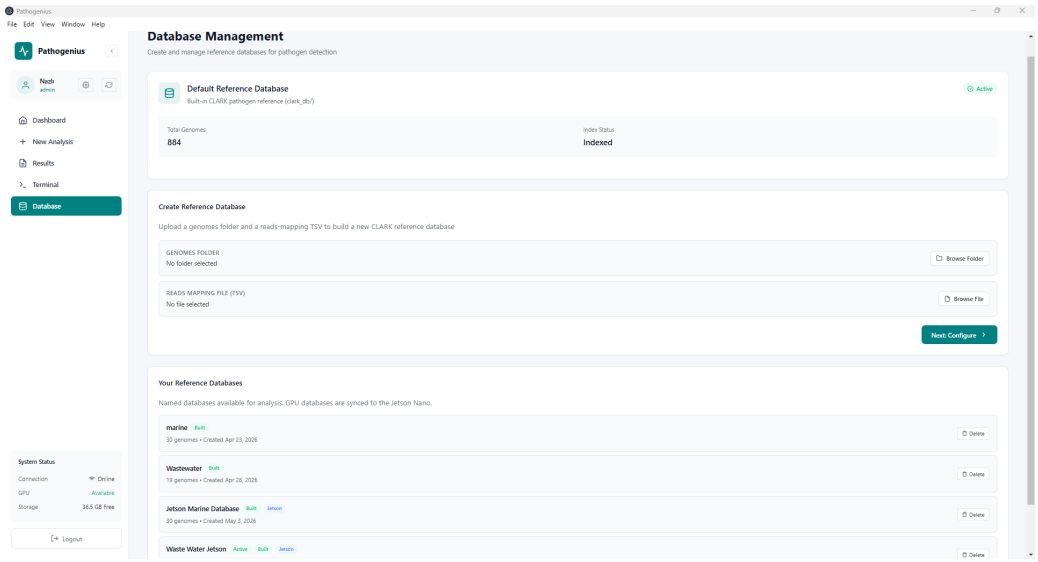

Figure 8.1: Default database card, custom database creation form, and database list.

The Database Management page allows you to view the built-in CLARK reference database and create custom databases from your own genome files.

8.2 Default Reference Database

A prominent card shows the built-in CLARK database:

- **Name:** Default Reference Database
- **Location:** `clark_db/`
- **Status:** Active
- **Statistics:** Total genome count and index status (*"Indexed ✔"* or *"Not built"*)

8.3 Creating a Custom Reference Database

Building a custom database is a two-step process within the page.

8.3.1 Step 1 — File Selection

1. Click [Browse] next to *Genomes Folder* and select the directory containing your FASTA genome files (`.fna`, `.fasta`, `.fa`, `.fsa`, or gzipped variants).
2. Click [Browse] next to *Reads Mapping File* and select the `reads_mapping.tsv` or `reads_mapping.tsv.gz` file (required for SPAdes-assembled contig taxid resolution).
3. Click [Next: Configure] (enabled only when both fields are populated).

8.3.2 Step 2 — Configuration

1. Enter a *Database Name* (auto-filled with the folder name as a suggestion).
2. Optionally enable the **Sync to Jetson Nano (GPU)** toggle to automatically transfer the finished database to the Jetson Nano. This is required to use the database with the GPU engine.
3. Click [Build Database].

A spinning *"Building database..."* indicator appears while the build runs. Building time depends on genome count and machine speed — typically a few minutes to several tens of minutes.

8.4 Managing Custom Databases

Built databases appear in the **Your Reference Databases** list. Each entry shows:

- Database name
- Status badges: **Active**, **Built** or **Not Built**, and **Jetson** if synced
- Genome count and creation date
- A Jetson sync warning if a sync is pending
- 🗑 [Delete] button — removes the database after a confirmation dialog

Chapter 9

Settings

9.1 Overview

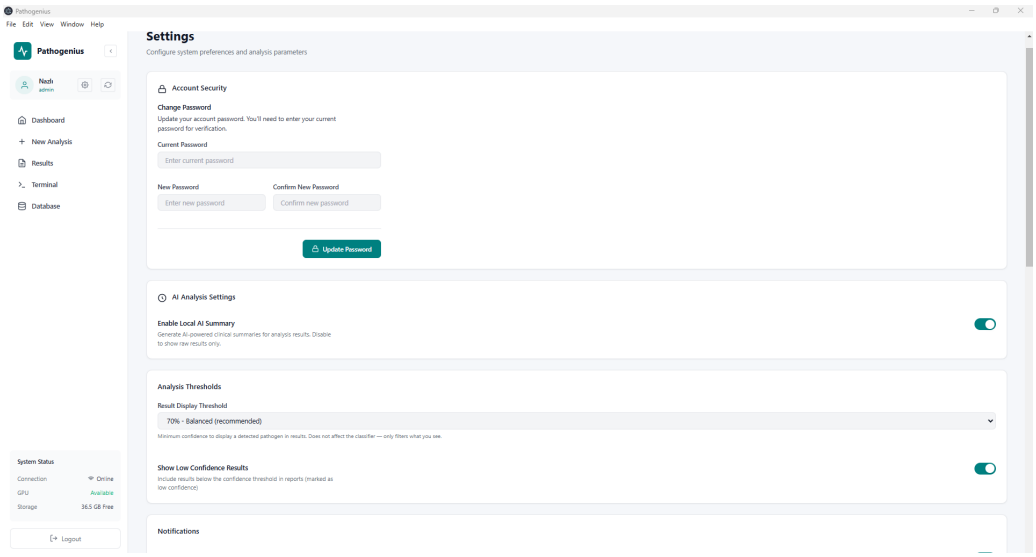

Figure 9.1: Settings page showing all configuration sections.

Access Settings from the ⚙ icon in the sidebar user card. Changes take effect immediately and are confirmed by a brief **Settings Saved** notification at the top of the page.

9.2 Account Security

> **Note**
>
> This section is hidden in Guest Mode.

9.2.1 Change Password

1. Enter your *Current Password*.
2. Enter a *New Password* (same requirements as registration: 12+ chars, uppercase, number, special character).

3. Enter the same password in *Confirm New Password*.
4. Click **[Update Password]**.

Error messages appear in red for mismatched passwords, incorrect current password, or passwords that do not meet complexity requirements. A green success message confirms the change.

9.3 AI Analysis Settings

- **Enable Local AI Summary** (toggle, default ON) — shows or hides the AI summary card on the Results page.

9.4 Analysis Thresholds

Setting	Description	Default
Confidence Threshold	Pathogens below this confidence score are excluded from reports. Options: 50% (very sensitive) to 90% (very precise).	70% (Balanced)
Minimum Read Count	Species with fewer than this many reads are excluded. Range: 10–10,000.	100
Show Low-Confidence Results	When ON, includes sub-threshold results in reports (shown with a warning label).	OFF

9.5 Data Management

- **Encrypt Local Data** (toggle, default OFF, hidden for guests) — encrypts stored analysis results using AES-256-GCM. A master key is stored in the OS keychain. Enabling this requires a short password-entry step on next startup.

9.6 Notifications

- **Analysis Complete Notifications** (toggle, default ON) — displays a desktop notification when an analysis finishes or fails.

9.7 Appearance

- **Dark Mode** (toggle, default OFF) — switches the application to a dark theme for reduced eye strain in low-light conditions.

9.8 Settings Import / Export / Reset

- **[Export Settings]** — saves all current settings as a JSON file for backup or transfer to another machine.
- **[Import Settings]** — loads a previously exported JSON settings file.
- **[Reset Settings]** — resets *all* settings to factory defaults. A confirmation dialog appears first.

9.9 Admin: User Management

> **Note**
>
> This section is visible only to accounts with the *Admin* role.

The User Management panel lists all registered accounts with columns for Username, Email, Role, Created Date, and Actions. Use the search box to filter users.

Available admin actions per user:

- **[Suspend / Activate]** — temporarily prevents or restores login access.
- **[Reset Password]** — sends a password reset email to the user.
- **[Make Admin / Remove Admin]** — promotes or demotes the user's role.

Actions on your own account are disabled (shown greyed out).

Chapter 10

Terminal

10.1 Overview

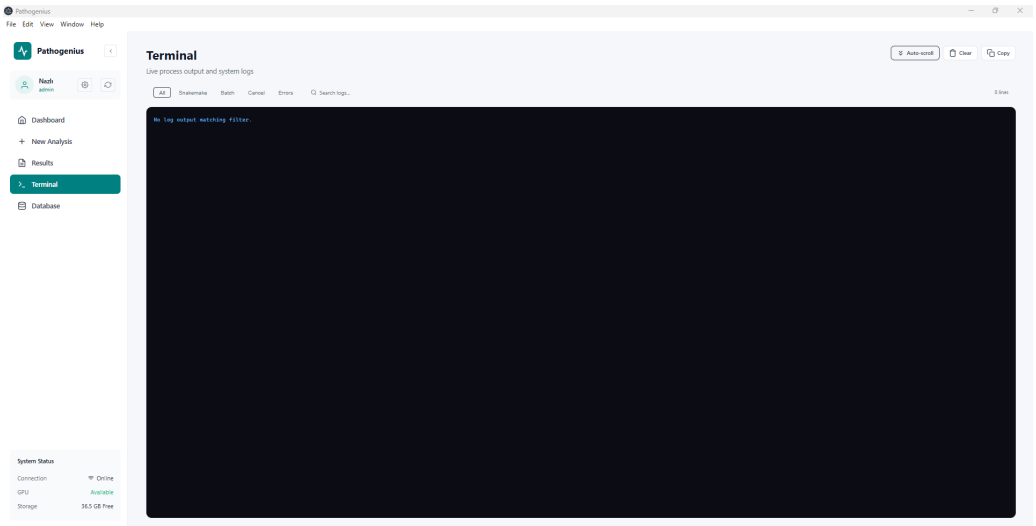

Figure 10.1: Embedded terminal window for direct pipeline interaction.

The Terminal page provides an embedded command-line interface for advanced users who need direct access to the underlying Snakemake pipeline and system shell.

10.2 Use Cases

- Running Snakemake manually with custom `-config` flags.
- Inspecting pipeline log files.
- Running `build_clark_db.py` with custom parameters.
- Debugging Docker or SSH connectivity issues.
- Executing arbitrary shell commands without leaving the application.

Warning

The embedded terminal runs with your system user privileges. Take care when executing commands that modify files, databases, or system configuration.

Chapter 11

GPU Mode (Jetson Nano)

11.1 Overview

GPU mode offloads the classification step to a Jetson Nano edge device running CU-CLARK-L, providing substantial speed improvements for large FASTQ files. Communication is secured via Tailscale VPN and SSH.

11.2 Prerequisites

Before using GPU mode:

1. Install and log in to **Tailscale** on both your workstation and the Jetson Nano.
2. Ensure the Jetson Nano is reachable at its Tailscale IP (verify with `ping <ip>`).
3. Configure SSH key-based authentication between your machine and the Jetson Nano (recommended for `ssh_batch_mode: true`).
4. Set up the CLARK database on the Jetson at the path specified in `config.yaml → jetson_nano.remote_db`.

11.3 Configuring the Connection

Edit `Patho-genius/config.yaml`:

```
jetson_nano:
  host: "100.71.242.74"       # Tailscale IP of the Jetson Nano
  user: "pathogen"            # SSH username
  ssh_batch_mode: true        # true = key auth; false = password prompts
  remote_db: "/home/pathogen/cuclark_db"
  remote_workspace: "/home/pathogen/pathogenius_ws"
  cuclark_dir: "/home/pathogen/cuclark"
```

11.4 GPU Pipeline Steps

When GPU engine is selected, the Snakemake pipeline performs the following operations automatically:

1. **Connectivity check** — SSH pre-check with a descriptive error if the Jetson is unreachable.
2. **FASTQ upload** — SCP transfer of the input file (skipped if the file already exists remotely).
3. **GPU memory cleanup** — drops page caches to free unified GPU memory.
4. **Stale result removal** — deletes previous `.clark.csv` to prevent false completion detection.
5. **CU-CLARK-L launch** — classification runs in background via `nohup` with `-b 128` batch size.
6. **Polling loop** — checks every 30 seconds: Running → DONE → result downloaded.
7. **CSV validation** — confirms the output file contains more than a header row.
8. **Result download** — SCP retrieves the classification CSV to the local machine.
9. **Remote cleanup** — removes intermediate files (FASTQ is retained for subsequent runs).
10. **Local abundance estimation** — reads NCBI taxonomy (cached after first download) to compute abundance percentages.

11.5 Syncing a Database to the Jetson Nano

When creating a custom database, enable the **Sync to Jetson Nano** toggle in the Database Management page. The database will be transferred to the Jetson after the build completes. Custom databases must be synced before they can be used in GPU mode.

Chapter 12

Cloud Synchronisation

12.1 Overview

Pathogenius integrates with Firebase Storage and Firestore to allow registered users to back up and share encrypted analysis results across devices.

> **Note**
>
> Cloud sync is **not available in Guest Mode**. You must be logged in with a registered account.

12.2 Uploading a Result

1. Navigate to the **Results** page.
2. Locate the completed analysis in the history table.
3. Click the [Upload] icon on the right side of the row.
4. The result is encrypted with AES-256-GCM using your account key and uploaded to Firebase Storage.
5. The row gains a cloud icon to indicate it is synchronised.

12.3 Downloading a Cloud Result

1. Switch to the **Cloud Results** tab on the Results page.
2. Click [Refresh] to fetch the latest list from Firebase.
3. Locate the analysis you wish to retrieve.
4. Click [Download].
5. The result is decrypted and stored locally, and its status changes to **Local copy**.

12.4 Encryption Details

All cloud-stored results are encrypted using:

- **Algorithm:** AES-256-GCM
- **Key storage:** OS keychain via `keytar` — the key never leaves your machine in plaintext
- **At rest:** Data is encrypted before leaving the application; Firebase never receives unencrypted content

Chapter 13

Sample Workflows

13.1 Scenario 1: Standard Clinical Sample Analysis

A researcher receives a FASTQ file from a clinical metagenomics run and needs to identify pathogens.

Step-by-Step Walkthrough

1. Launch Pathogenius and log in with your credentials.
2. Click **New Analysis** in the sidebar.
3. Drag the FASTQ file onto the upload zone. Click **[Next: Configuration]**.
4. Select **CPU Workflow** (ensure Docker is running). Set the Analysis Name to the patient ID code. Set Sample Type to *Clinical Sample*. Leave Confidence Threshold at 70%. Click **[Next: Launch]**.
5. Review the summary. Click **[Start Analysis]**.
6. Monitor progress on the Dashboard. Expect 15–30 minutes for a typical dataset.
7. When complete, navigate to **Results**, find the analysis, and click **[View Report]**.
8. Review pathogen cards and risk levels. Click **[Generate Summary]** for the AI clinical interpretation.
9. Click **[Export Report]** to save a PDF for the patient record.
10. Optionally, click the upload icon to back up the result to cloud storage.

13.2 Scenario 2: High-Throughput Field Analysis with GPU

A field worker has a large FASTQ dataset and a Jetson Nano device available.

Step-by-Step Walkthrough

1. Ensure Tailscale is running on both the workstation and the Jetson Nano.
2. Launch Pathogenius. Click **New Analysis**.
3. Select the large FASTQ file. If a *"Large file detected"* warning appears, enable **Batch Processing**.
4. Click **[Next: Configuration]**. Select **GPU Accelerated**.

5. Name the analysis, choose the appropriate database (ensure it is synced to the Jetson), and set confidence threshold. Click [Next: Launch].
6. Review and click [Start Analysis].
7. The pipeline uploads the FASTQ to the Jetson, runs CU-CLARK-L, polls for completion, and downloads results automatically.
8. View results as in Scenario 1.

13.3 Scenario 3: Building a Custom Pathogen Database

A researcher has assembled novel pathogen genomes and wants to add them to Pathogenius.

Step-by-Step Walkthrough

1. Place genome FASTA files and the reads mapping TSV in a dedicated folder.
2. In Pathogenius, navigate to **Database** in the sidebar.
3. Under *Create Reference Database*, click [Browse] for *Genomes Folder* and select the folder.
4. Click [Browse] for *Reads Mapping File* and select the `.tsv` or `.tsv.gz` file.
5. Click [Next: Configure]. Enter a descriptive name.
6. If GPU analyses are planned, enable **Sync to Jetson Nano**.
7. Click [Build Database] and wait for the build to complete.
8. The new database now appears in *Your Reference Databases* and in the database dropdown when creating a new analysis.

13.4 Scenario 4: Multi-User Lab Setup with Admin Controls

A lab administrator manages multiple researcher accounts.

Step-by-Step Walkthrough

1. Log in with an admin account. Navigate to **Settings**.
2. Scroll to the **Admin: User Management** section.
3. Use the search box to find a specific user.
4. To grant admin access, click [Make Admin] on that user's row.
5. To temporarily block a user, click [Suspend]; they will see an error on their next login.
6. To help a user who cannot log in, click [Reset Password] to send them a reset email.

Chapter 14

Troubleshooting

Problem	Solution
`npm start` fails with *"electron not found"*	Run `npm install` inside the `frontend/` directory first.
"Workflow directory not found"	Ensure the `Patho-genius/` directory exists alongside `frontend/`.
Analysis stuck at *"Starting..."*	Check that Docker Desktop is running (CPU mode) or Tailscale is connected and the Jetson Nano is reachable (GPU mode).
"Could not start Snakemake"	Install Python 3 and run `pip install snakemake`.
Database build fails at NCBI lookup	Check your internet connection. NCBI rate-limits requests — wait a few minutes and retry.
`build_clark_db.py` — *"No genome_sources defined"*	Add entries under `genome_sources` in `config.yaml`.
100% unclassified reads	Rebuild the database: delete `clark_db/Custom/` and `targets.txt`, then rerun `build_clark_db.py`.
Docker permission errors	Enable file sharing for the workspace directory in Docker Desktop settings.
GPU mode — *"Cannot reach Jetson Nano"*	Verify Tailscale VPN is active on both machines. If using password auth, set `ssh_batch_mode: false` in `config.yaml`.
GPU — header-only CSV / 0 taxa	CUDA watchdog timeout. Ensure `-b 128` is in the CU-CLARK-L command. Alternatively, disable the watchdog: `echo 0 | sudo tee /sys/kernel/debug/gpu.0/timeouts_enabled`.
GPU — poll loop stuck on RUNNING	Stale process detection. Ensure the poll command uses `pgrep -f '[c]uCLARK-l'` (bracket trick).
GPU — *"Nothing to be done"*	Stale cached results. The analysis service auto-cleans `.clark.csv`, `.abundance.csv`, and `.json` before each run. If running standalone, add `-rerun-incomplete`.
LLM — *"Failed to load model"*	Ensure Node.js 22+ is installed. Check that a `.gguf` file exists in `frontend/models/`.

Problem	Solution
LLM — *"Unexpected token 'with'"*	Upgrade Node.js to v22+ and reinstall: `rm -rf node_modules && npm install`.
Blank Electron window on startup	Open the DevTools console: uncomment `mainWindow.webContents.openDevTools()` in `main.js`, restart, and check the console for errors.
Cloud sync — *"Not authenticated"*	Log in with a registered account. Cloud features are unavailable in Guest Mode.
Email verification email not received	Check spam/junk folders. Click **[Resend Verification Email]** on the login page.
Settings not saving	Ensure you have write permissions to the `frontend/` directory. Restart the application and try again.

Chapter 15

Frequently Asked Questions

Q: Can I use Pathogenius without Docker?
Docker is required for CPU-mode classification (CLARK-l runs inside a container). If Docker is unavailable, you can use GPU mode via the Jetson Nano instead. Building the reference database also requires Docker.

Q: How long does an analysis typically take?
A typical clinical FASTQ file (300 MB–2 GB) takes 15–30 minutes in CPU mode on an 8-core machine. GPU mode on a Jetson Nano is 3–5× faster for large files. Very large files (10+ GB) may take over an hour in CPU mode.

Q: Is my patient data sent to the cloud?
Only if you explicitly upload results using the cloud sync feature. All classification runs on your local machine. The AI summary also runs locally. If you do upload, data is encrypted with AES-256-GCM before leaving the application.

Q: What FASTQ formats are supported?
Standard FASTQ (`.fastq`) and gzip-compressed FASTQ (`.fastq.gz`). Both single-end and paired-end reads are accepted; however, the pipeline currently processes a single file per analysis.

Q: How do I update the pathogen reference database?
Add new genome FASTA files to your genome source directory and rebuild using `build_clark_db.py` (via the Terminal page or command line). Alternatively, use the Database Management page to create a new custom database from the updated files.

Q: What does the confidence threshold control?
The confidence threshold is the minimum score assigned by CLARK-l to accept a species identification. A higher threshold (e.g., 90%) reduces false positives but may miss low-abundance pathogens. A lower threshold (e.g., 50%) increases sensitivity but may produce more noise.

Q: Can multiple users share the same installation?
Yes. Each user has a separate account and their results are stored under their user profile. Admin accounts can manage all users via the Settings page.

Q: The AI summary card is not visible. What should I do?
Ensure **Enable Local AI Summary** is toggled ON in Settings. Also verify that a `.gguf` model file exists in `frontend/models/`. The model status is displayed in the AI summary card header.

Q: How do I run Pathogenius in development/demo mode without real data?
Set the environment variable `PATHOGENIUS_MOCK_ANALYSIS=1` before launching. On Windows: `set PATHOGENIUS_MOCK_ANALYSIS=1 && npm start`. This returns synthetic results after a simulated delay, allowing UI testing without Docker or a database.

Q: Can I export results to share with a colleague?
Yes — two options are available from the detail view: **[Export Report]** generates a formatted PDF, and **[View Raw Data]** shows the JSON output which can be saved manually. Registered users can also share results via cloud sync.

Chapter 16

Appendix

16.1 Output JSON Format

Each completed analysis produces a JSON file with the following structure:

```
{
  "analysis_name": "Patient_001_Sample",
  "sample_type":   "clinical",
  "completed_at":  "2026-04-25T20:20:00.000Z",
  "classifier":    "CU-CLARK-L",
  "summary": {
    "total_reads":         908893,
    "classified_reads":    14585,
    "classification_rate": 1.6,
    "species_detected":    23,
    "pathogens_detected":  3
  },
  "pathogens": [
    {
      "name":             "Escherichia coli",
      "strain":           "Escherichia coli",
      "tax_id":           562,
      "abundance":        45.2,
      "reads":            6594,
      "confidence":       95,
      "risk_level":       "high",
      "amr_genes":        5,
      "virulence_genes":  12
    }
  ],
  "taxonomy": {
    "bacteria":       63.6,
    "viruses":         4.8,
    "proteobacteria": 48.0,
    "firmicutes":     18.0
  }
}
```

16.2 Configuration Reference (`config.yaml`)

Key	Default	Description
`engine`	`"cpu"`	Active classification engine: `"cpu"` or `"gpu"`
`paths.db_host_windows`	`"./clark_db"`	Path to the CLARK reference database
`paths.data_host_windows`	`"./fastQ_reads"`	Directory for input FASTQ files
`sample`	`"SRR7497167_1"`	Default sample basename
`clark.image`	(CLARK image tag)	Docker image for CPU classification
`clark.threads`	8	Classification threads
`clark.kmer_length`	27	k-mer length
`clark.sampling_factor`	2	Sampling factor (higher = faster, less sensitive)
`jetson_nano.host`	—	Tailscale IP of the Jetson Nano
`jetson_nano.user`	—	SSH username on the Jetson
`jetson_nano.ssh_batch_mode`	`true`	`true` = key auth, `false` = password auth
`jetson_nano.remote_db`	—	Database path on the Jetson
`jetson_nano.remote_workspace`		Working directory on the Jetson
`jetson_nano.cuclark_dir`	—	CU-CLARK-L installation path on the Jetson

16.3 Environment Variables

Variable	Description	Default
`SNAKEMAKE_WORKFLOW_DIR`	Path to the Snakemake workflow directory	`<repo>/Patho-genius`
`RESULTS_DIR`	Output directory for analysis JSON files	`<repo>/frontend/results`
`PATHOGENIUS_MOCK_ANALYSIS`	Set to `1` to return synthetic results	`0`

16.4 Keyboard Shortcuts & Quick Reference

Action	Shortcut / Location
Open Dashboard	Click in the sidebar
Start New Analysis	Click in the sidebar
View Results	Click in the sidebar
Open Terminal	Click in the sidebar
Open Database Manager	Click in the sidebar
Open Settings	Click in the user card
Logout	Click Logout at sidebar bottom
Refresh System Status	Click in the user card

Disclaimer

Pathogenius is provided as a research and decision-support tool. The results produced by the classification pipeline are computational predictions based on sequence similarity to reference databases and are subject to the quality of those databases and the input data.

Pathogenius results must not be used as the sole basis for clinical diagnosis, patient treatment decisions, or public health interventions. All findings should be confirmed using validated clinical laboratory methods by qualified healthcare professionals.

The Pathogenius Team makes no warranty, express or implied, regarding the accuracy, completeness, or fitness for any particular purpose of the results generated by this software.