GrantCore.ai

AI-Powered Grant Management Platform

Administrator Guide

System Administration, Configuration & Security
for Platform Administrators

Version 1.0 | March 2026
SoftSim Technologies Inc. | Montreal, Canada
https://grantcore.ai

Table of Contents

PART

Getting Started

CHAPTER 1

Administrator Overview

This guide is intended for GrantCore.ai platform administrators. Administrators have full access to all researcher features plus exclusive access to system configuration, user management, funder database administration, AI provider configuration, and security audit capabilities.

Administrator Responsibilities

- Manage user accounts — create, activate, deactivate, and delete researcher and admin accounts
- Configure AI provider — set up system-wide AI credentials (Claude, OpenAI, or Gemini) for all users
- Manage funders — add, edit, and remove funding organizations and their programs
- Monitor security — review login audit trails, identify suspicious access patterns, and enforce account policies
- System maintenance — run database migrations, deploy updates, and monitor server health

Admin Navigation

When logged in as an administrator, the sidebar navigation shows additional links under the "ADMIN" section:

- Admin: Users — User account management (admin_users.html)
- Admin: Settings — System-wide AI configuration (admin_settings.html)
- Admin: Funders — Funder database management (admin_funders.html)
- Admin: Audit — Login audit trail and security monitoring (admin_audit.html)

These links are hidden from non-admin users. The admin section appears below the standard workflow and tools sections in the sidebar.

Prerequisites

- An account with the ADMIN role (the first admin account is typically created during initial deployment)
- Access to the GrantCore.ai web application at your organization’s URL
- For AI configuration: an API key from your chosen AI provider (Claude, OpenAI, or Gemini)
- For server administration: SSH access to the production server (if self-hosted)

CHAPTER 2

First-Time Setup

After deploying GrantCore.ai for the first time, follow these steps to configure the platform for your organization.

Step 1: Log In as Admin

Navigate to your GrantCore.ai URL and log in with the default administrator credentials. The initial admin account is created during database migration with email admin@grantcore.ai.

WARNING
Change the default admin password immediately after first login. Navigate to your profile settings or use the Admin: Users page to reset the password to a strong, unique value.

Step 2: Configure AI Provider

1. Click "Admin: Settings" in the sidebar.
2. Select your preferred AI provider: Claude (recommended), OpenAI, or Gemini.
3. Enter the API key obtained from your provider's developer console.
4. Click "Save AI Settings" to persist the configuration.
5. Click "Test Connection" to verify the AI responds correctly.

Step 3: Add Funders

Five funder packs are pre-installed (CIHR, NSERC, NRC, NSF, NIH). To add additional funders:

1. Click "Admin: Funders" in the sidebar.
2. Select the jurisdiction/country from the dropdown.
3. Enter the funder name and optional website URL.
4. Click "Add Funder" to create the record.

Step 4: Create User Accounts

1. Click "Admin: Users" in the sidebar.
2. Fill in the user's full name, email, password, and role.
3. Click "Create User" to create the account.
4. Optionally send an activation code for controlled onboarding.

Share the login credentials with each user securely. Users can change their password after first login via the Settings page.

TIP

We recommend configuring the AI provider before creating user accounts, so researchers can use AI features immediately upon first login.

PART

User Management

CHAPTER 3

Managing User Accounts

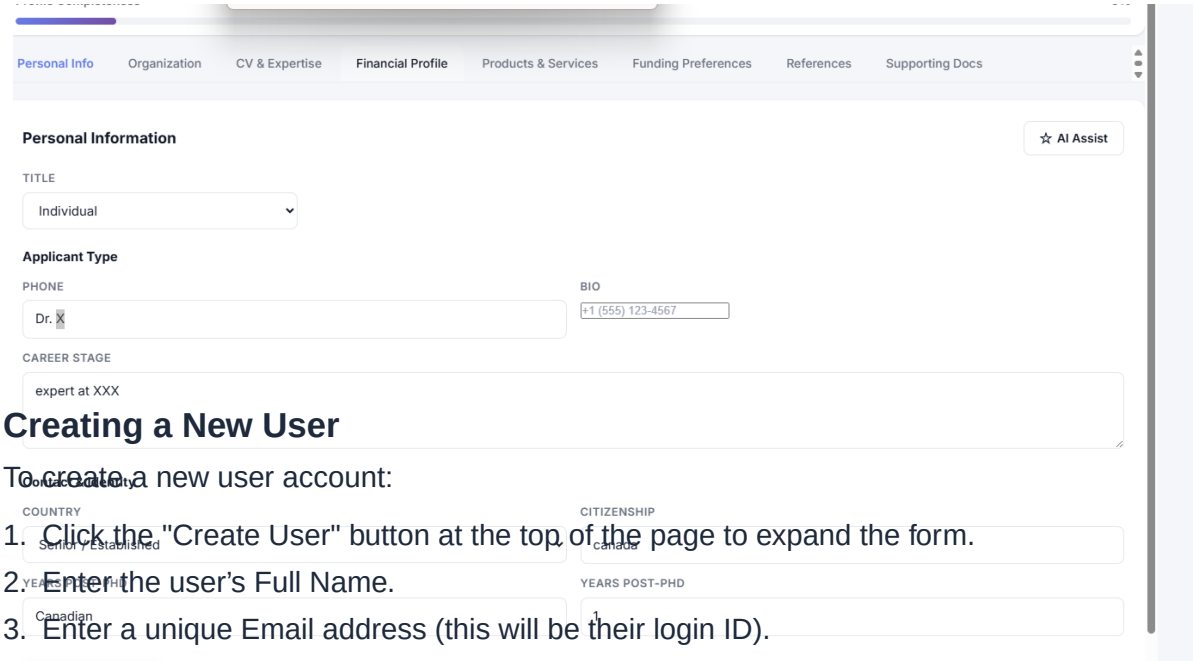
The Admin: Users page (admin_users.html) is the central hub for managing all user accounts. Only administrators can access this page.

Users Table

The main table displays all registered users with the following columns:

- ID — Unique numeric identifier for each user
- Name — User's full name
- Email — Login email address (must be unique across the system)
- Role — RESEARCHER (default) or ADMIN, shown as color-coded badges
- Status — Active (green badge) or Pending (orange badge)
- Created — Account creation date
- Actions — Context-sensitive action buttons

Figure: Capability Profile page showing form fields and tabs (Windows)



Creating a New User

To create a new user account:

1. Click the "Create User" button at the top of the page to expand the form.
2. Enter the user's Full Name.
3. Enter a unique Email address (this will be their login ID).
4. Set an initial Password (minimum 8 characters). Passwords are hashed with bcrypt (12 salt rounds).
5. Select a Role: Researcher (default) or Admin.
6. Click "Create User" to save.

The system automatically creates a personal workspace for the new user. They can log in immediately with the provided credentials.

DUPLICATE EMAIL
If the email address is already in use, the system returns a 409 Conflict error. Each email address can only be associated with one account.

User Roles

GrantCore.ai uses two roles:

Researcher

The default role for all new accounts. Researchers have full access to the 8-step grant workflow, AI features, CV management, post-award management, and their own settings. They cannot access admin pages or manage other users.

Administrator

Administrators have all researcher capabilities plus exclusive access to:

- User management — create, edit, activate, deactivate, and delete accounts
- Funder management — add and remove funders from the database
- AI configuration — set system-wide AI provider and API key
- Audit trail — view login history and security metrics for all users

To change a user's role, find them in the users table and click the role toggle action. The change takes effect on their next login.

CHAPTER 4

Activation & Onboarding

GrantCore.ai supports a flexible activation workflow for controlled user onboarding.

Direct Activation

By default, new accounts created by an administrator are immediately active. Users can log in with their email and password right away.

Activation Code Workflow

For organizations requiring controlled onboarding, administrators can generate activation codes:

1. Find the user in the users table (status will show "Pending" if not yet activated).
2. Click the "Send Activation Code" button.
3. Select the code validity period from the dropdown:

- 1 Day — Code expires after 24 hours
- 3 Days — Short-term access
- 7 Days — Standard onboarding period
- 30 Days — Extended onboarding
- 1 Year — Long-term validity
- Permanent — Code never expires

1. Click "Send" to generate and email the code.
2. The system generates a 6-character hexadecimal code (e.g., "A3F29D").
3. An HTML email is sent to the user containing the activation code, login link, and expiry information.
4. A BCC copy is sent to admin@grantcore.ai for record-keeping.

Activation Email

The activation email includes:

- A styled header with the GrantCore.ai brand
- The 6-character activation code displayed prominently (32px font)
- A direct link to the activation page
- The code expiry date (or "never expires" for permanent codes)
- Login instructions with the application URL

Activate Directly

Administrators can also activate a pending account without sending an email by clicking "Activate Directly" on the user's row. This immediately sets the account status to Active.

CHAPTER 5

Account Management Actions

Each user row in the table provides context-sensitive action buttons based on the user's current status.

Reset Password

Administrators can reset any user's password:

1. Click "Reset Password" on the user's row.
2. Enter the new password in the modal dialog (minimum 8 characters).
3. Click "Reset" to save the new password.

The new password is hashed with bcrypt (12 salt rounds) and stored securely. The user must use the new password on their next login. There is no email notification for password resets—communicate the new password to the user securely.

Enable / Disable Account

Toggle a user's active status:

- Active !' Disabled: Click "Deactivate" to prevent the user from logging in. Their data is preserved but they cannot access the system.
- Disabled !' Active: Click "Activate" to re-enable login access.

Disabled accounts remain in the database with all their data intact. This is useful for temporary access suspension without data loss.

Delete User

Permanently remove a user account:

1. Click "Delete" on the user's row.
2. Confirm the deletion in the dialog.

WARNING

Deleting a user permanently removes their account, activation codes, and workspace membership records. This action cannot be undone. The user's workspace and its contents (applications, documents, CV data) are also affected. Administrators cannot delete their own account to prevent accidental lockout.

Change Role

To promote a researcher to admin or demote an admin to researcher, use the PATCH endpoint or update the role field in the users table. The change takes effect on the user’s next login or API request.

PART

AI Configuration

CHAPTER 6

System AI Configuration

The Admin: Settings page (admin_settings.html) allows administrators to configure a system-wide AI provider and API key. These settings serve as the default for all users who have not configured their own personal AI key.

Accessing the Settings Page

1. Log in with an Administrator account.
2. Click "Admin: Settings" in the sidebar navigation.
3. The page displays an "AI Configuration" card with provider selection, API key input, and system status.

Supported AI Providers

Three providers are supported, each with different strengths:

Claude (Anthropic) — Recommended

- Best for academic writing, peer review simulation, and nuanced eligibility assessment
- Default model: claude-sonnet-4-5-20250929
- API key console: console.anthropic.com
- Pricing: Pay-per-token with generous rate limits

OpenAI (GPT-4o)

- Strong general-purpose AI with wide model selection
- Default model: gpt-4o
- API key console: platform.openai.com
- Good alternative if Claude is unavailable in your region

Gemini (Google)

- Fast and cost-effective for high-volume usage
- Default model: gemini-2.5-flash
- API key console: aistudio.google.com
- Good for organizations already using Google Cloud

Configuration Steps

1. Select your provider by clicking the radio card (Claude, OpenAI, or Gemini). The selection highlights in blue.

2. Enter your API key in the "API Key" field. A dynamic hint below the field shows the specific console URL for your selected provider.

3. Optionally enter a Model Override to use a specific model variant (e.g., "claude-opus-4-20250918" or "gpt-4-turbo").

4. Click "Save AI Settings" to persist the configuration to the database.

5. Click "Test Connection" to verify the AI responds correctly. A success message appears in green; failure in red.

API Key Security

- API keys are stored in the system_settings database table.
- When retrieved via the GET endpoint, keys are masked: only the last 4 characters are visible (e.g., "***abcd").
- The key is never exposed in the frontend after saving—only the masked version is shown.
- When saving a new key, the full key is transmitted over HTTPS and stored server-side.
- Runtime environment variables are also updated immediately so changes take effect without server restart.

System Status Card

Below the configuration form, the System Status card displays:

- Provider — The currently active AI provider name
- API Key — Masked key status (e.g., "***a1b2" or "Not configured")
- Model — The active model name (or "default" if no override)

This card automatically updates after saving settings, providing visual confirmation.

CHAPTER 7

AI Settings Fallback Chain

GrantCore.ai uses a three-level fallback chain to determine which AI credentials to use for any given user:

Level 1: User Settings (Highest Priority)

Individual users can configure their own AI provider and API key via the Settings page (settings.html). If a user has saved a valid provider and API key, their personal settings are used for all AI operations.

Use case: A researcher who wants to use their own OpenAI key while the system default is Claude.

Level 2: System Settings (Admin-Configured)

If a user has not configured personal AI settings, the system falls back to the admin-configured settings from the Admin: Settings page. This is the most common configuration—the admin sets up one API key and all users share it.

Use case: A university administrator configures a single Claude API key for the entire research team.

Level 3: Environment Variables (Lowest Priority)

If neither user nor system settings are configured, the server falls back to environment variables:

- AI_PROVIDER — Provider name (claude, openai, gemini)
- ANTHROPIC_API_KEY — Claude API key
- OPENAI_API_KEY — OpenAI API key
- GEMINI_API_KEY — Gemini API key
- AI_MODEL — Optional model override

These are set in the server's .env file and are useful for development or as a final fallback.

TIP

For most deployments, we recommend using Level 2 (System Settings). Configure the AI key once in Admin: Settings and all users benefit automatically. Reserve Level 1 for power users who want their own provider.

How the Fallback Works (Technical)

When an AI operation is triggered, the backend calls getUserAiSettings(userId):

1. Query user_settings table for the user’s personal ai_provider and ai_api_key.
2. If both are present, return the user’s settings.
3. Otherwise, query system_settings table for system-wide ai_* keys.
4. Merge system settings with any partial user settings (user settings override).
5. If the merged result still lacks credentials, fall back to process.env variables.
6. Return the final merged settings object to the AI service.

PART

Funder Management

CHAPTER 8

Funder Database

The Admin: Funders page (admin_funders.html) allows administrators to manage the funder database. Funders are the organizations that provide grant funding. Each funder belongs to a jurisdiction and can have multiple programs.

Viewing Funders

The funders table shows all registered funding organizations with:

- ID — Unique numeric identifier
- Country/Jurisdiction — The funder's home jurisdiction
- Name — Full funder name
- Website — Clickable link to the funder's website (if provided)
- Actions — Delete button

Filtering & Search

Use the filter bar above the table:

- Jurisdiction dropdown — Filter by country: All, Canada, US, EU, UK, France, Germany, Australia, New Zealand, Other
- Search input — Search by funder name or website URL (case-insensitive substring match)
- Funder count — Displays the total number of funders matching the current filters

Adding a Funder

1. Click the form at the top of the page to expand it.
2. Select the Country/Jurisdiction from the dropdown.
3. Enter the Funder Name (must be unique in the database).
4. Enter the Website URL (optional but recommended).
5. Click "Add Funder" to save.

The new funder appears immediately in the table and becomes available in the application creation workflow for researchers.

Deleting a Funder

Click the "Delete" button on any funder row. A confirmation dialog warns about cascading effects on linked applications. After confirmation, the funder and all its programs are removed from the database.

WARNING

Deleting a funder also affects any applications linked to that funder. Researchers will see an "Unknown Funder" label on affected applications. Consider deactivating instead of deleting if applications exist.

CHAPTER 9

Built-in Funder Packs

GrantCore.ai includes five pre-installed funder packs with comprehensive program data, compliance rules, evaluation criteria, and portal submission templates.

CIHR — Canadian Institutes of Health Research

- Programs: Project Grant (max 75 pages, $50K–$750K), Catalyst Grant
- Compliance rules: CCV required, institutional signature, ethics approval
- Portal template: ResearchNet step-by-step field mapping
- Evaluation criteria: Scientific excellence, feasibility, impact, SGBA+

NSERC — Natural Sciences & Engineering Research Council

- Programs: Discovery Grant, Alliance Grant
- Compliance rules: CCV required, 5-page limit (Discovery)
- Portal template: NSERC online system field mapping
- Evaluation criteria: Discovery potential, training of HQP, significance

NRC-IRAP — National Research Council

- Programs: Industrial Research Assistance Program
- Focus: SME innovation, technology readiness, commercialization
- Compliance rules: Canadian SME, <500 employees
- Evaluation criteria: Innovation, business plan, team, market potential

NSF — National Science Foundation (US)

- Programs: Standard Grant (15 pages), CAREER Award
- Compliance rules: NSF Biographical Sketch required, budget justification
- Portal template: Research.gov submission guide
- Evaluation criteria: Intellectual merit, broader impacts

NIH — National Institutes of Health (US)

- Programs: R01 Research Project Grant (12 pages), R21 Exploratory Grant
- Compliance rules: NIH Biosketch required, specific aims page
- Portal template: eRA Commons step-by-step guide
- Evaluation criteria: Significance, investigators, innovation, approach, environment

Additional Canadian Funders

The following funders are included with basic program information and can be enhanced by administrators:

- SSHRC, Mitacs, CFI, Genome Canada, CANARIE
- Canada Council for the Arts, Brain Canada Foundation
- Heart and Stroke Foundation, Canadian Cancer Society
- Terry Fox Research Institute, Ontario Research Fund
- Fonds de recherche du Québec (FRQ), Michael Smith Foundation
- Alberta Innovates, Saskatchewan Health Research Foundation
- Research Manitoba, New Brunswick Innovation Foundation, SickKids Foundation

PART

Security & Audit

CHAPTER 10

Login Audit Trail

The Admin: Audit page (admin_audit.html) provides comprehensive logging of all authentication attempts. Every login—successful or failed—is recorded with detailed metadata for security monitoring.

Statistics Dashboard

Four metric cards at the top of the page provide at-a-glance security insights:

- Last 24 Hours — Total login attempts (all types) in the past day
- Successful (24h) — Successful login count with green indicator
- Failed (24h) — Failed login attempts with red indicator
- Failed (7 Days) — Weekly failed attempt trend with red indicator

These statistics auto-refresh when the page loads and after applying filters.

Audit Log Table

The detailed table logs every authentication event with:

- Time — Timestamp formatted as YYYY-MM-DD HH:MM
- Email — The email address used in the login attempt
- User Name — The matched user's name (or "-" if email not found in system)
- Status — Success (green badge) or Failed (red badge)
- IP Address — Source IP address of the request (shown in smaller muted text)
- Failure Reason — Why the login failed (shown in smaller muted text)

Common Failure Reasons

- "Invalid email" — No account exists with the attempted email address
- "Invalid password" — Correct email but wrong password
- "Account disabled" — The account exists but is deactivated by an admin
- "Account not activated" — The account requires activation code entry

Filtering the Audit Log

Use the filter controls above the table:

- Email filter — Search by email address (partial match with ILIKE, e.g., "john" matches "john@example.com")

- Status dropdown — All, Successful, or Failed
- Time period — Last 24 hours, Last 7 days, Last 30 days, Last 90 days, or All time

Click "Search" or press Enter to apply filters. Results are limited to 200 entries by default (maximum 1000).

Security Best Practices

TIP

Review the audit trail at least weekly. Look for: (1) Multiple failed attempts from the same IP address, which may indicate a brute-force attack. (2) Failed attempts targeting the same email, suggesting credential guessing. (3) Successful logins from unusual IP addresses or geographic regions. (4) Login activity outside normal business hours.

CHAPTER 11

Security Architecture

Understanding the security measures built into GrantCore.ai helps administrators make informed decisions about configuration and monitoring.

Authentication

- JWT tokens with 7-day expiry (configurable via JWT_EXPIRES_IN environment variable)
- HS256 signing algorithm with a required secret key (JWT_SECRET in .env)
- Tokens are sent as HTTP-only cookies or Authorization Bearer headers
- Password hashing with bcrypt using 12 salt rounds (computationally expensive to brute-force)
- JWT_REQUIRED=true enforces token validation on all protected routes

Authorization

- Role-based access control: RESEARCHER and ADMIN roles
- requireAuth middleware validates JWT token on every request
- requireRole("ADMIN") middleware restricts admin routes
- verifyOwner middleware factory ensures workspace-scoped data isolation
- Users can only access data within their own workspace

HTTP Security Headers

Helmet.js is configured with the following security headers:

- Content-Security-Policy — Restricts resource loading origins
- X-Content-Type-Options: nosniff — Prevents MIME type sniffing
- X-Frame-Options: DENY — Prevents clickjacking via iframes
- Strict-Transport-Security — Enforces HTTPS connections
- X-XSS-Protection — Enables browser XSS filtering

Rate Limiting

Built-in rate limiting protects against brute-force attacks:

- Login endpoint: 20 attempts per 15-minute window per IP
- Registration endpoint: 10 attempts per 60-minute window per IP
- AI endpoints: Rate-limited to prevent API key abuse

When a rate limit is exceeded, the server responds with HTTP 429 Too Many Requests.

Data Isolation

GrantCore.ai uses workspace-based data isolation:

- Each user has a personal workspace created at account creation time
- All user data (applications, documents, CV entries, awards) is scoped to their workspace
- The verifyOwner middleware ensures users cannot access other workspaces' data
- Administrators can view all user accounts but not their workspace data directly

Input Validation

- All user inputs are validated server-side before database queries
- Parameterized SQL queries prevent SQL injection
- HTML output is escaped via escHtml() utility to prevent XSS
- File uploads are restricted to allowed MIME types (PDF, DOCX, DOC, TXT)
- Maximum file size: 10MB per upload

PART

Server Administration

CHAPTER 12

Deployment & Infrastructure

This chapter covers the server infrastructure for self-hosted GrantCore.ai deployments.

System Requirements

- Node.js 18+ (LTS recommended)
- PostgreSQL 14+
- Nginx (reverse proxy)
- Linux server (Ubuntu 22.04 LTS recommended)
- Minimum 2GB RAM, 20GB disk
- HTTPS certificate (Let's Encrypt recommended)

Directory Structure

On the production server:

- /var/www/grantcore-ai/ — Application root
- /var/www/grantcore-ai/backend/ — Express.js backend and API
- /var/www/grantcore-ai/frontend/ — Static HTML/CSS/JS frontend
- /var/www/grantcore-ai/backend/.env — Environment configuration (not in version control)
- /var/www/grantcore-ai/backend/uploads/ — User-uploaded files

Environment Variables

Key environment variables in backend/.env:

Variable	Description
PORT	Server port (default: 3009)
DATABASE_URL	PostgreSQL connection string
JWT_SECRET	Secret key for JWT signing (required in production)
JWT_EXPIRES_IN	Token expiry (default: 7d)
JWT_REQUIRED	Enforce JWT validation (set to true)
AI_PROVIDER	Default AI provider (claude/openai/gemini)
ANTHROPIC_API_KEY	Claude API key (fallback)
OPENAI_API_KEY	OpenAI API key (fallback)
GEMINI_API_KEY	Gemini API key (fallback)
NODE_ENV	Environment (production/development)
CORS_ORIGIN	Allowed CORS origins
SMTP_HOST	Mail server for activation emails

SMTP_PORT	Mail server port

Deployment Script

The included deployment script (scripts/deploy-axcess1.sh) automates the deployment process:

1. rsync — Syncs code to the production server (excludes node_modules, uploads, .env)
2. npm install --omit=dev — Installs production dependencies on the server
3. npm run migrate — Runs any pending database migrations
4. Nginx config — Copies and reloads the Nginx reverse proxy configuration
5. systemctl restart — Restarts the GrantCore.ai systemd service

Customize the script for your server hostname, user, and paths.

CHAPTER 13

Database Administration

GrantCore.ai uses PostgreSQL for all data storage.

Database Schema

The database contains 25+ tables organized by feature area:

Core Tables

- users — User accounts with email, password hash, role, and status
- workspaces — Workspace containers for data isolation
- workspace_users — User-to-workspace membership mapping

Grant Workflow Tables

- funders — Funding organizations with jurisdiction and website
- opportunities — Programs/opportunities within funders
- applications — Grant applications with status, amount, and metadata
- documents — Uploaded files with extracted text
- review_findings — AI-extracted findings from documents
- issues — Tracked issues from gap analysis
- doc_patches — Document-level edit suggestions
- checks — Compliance check results

CCV Tables

- ccv_profiles — Canadian Common CV profile data
- ccv_entries — Individual CV entries across 10+ sections

Post-Award Tables

- grant_awards — Awarded grants with amount, period, and status
- award_milestones — Project milestones with target dates
- award_deliverables — Required outputs linked to milestones
- award_budget — Budget line items by category
- award_claims — Expense reimbursement claims
- award_tasks — Project tasks with priority and status
- award_reports — Progress and final reports
- application_collaborators — Team member access with roles and permissions

Administration Tables

- system_settings — Admin-configured settings (AI provider, API keys)
- activation_codes — User activation codes with expiry
- login_audit — Authentication event log
- user_settings — Per-user preferences and AI configuration
- funder_configs — Funder-specific configuration overrides
- submission_records — Application submission history
- grants_migrations — Migration tracking table

Running Migrations

Database schema changes are managed through sequential numbered SQL migration files:

1. Migration files are stored in backend/src/grants/db/migrations/ (001 through 030+).
2. Run migrations with: node backend/src/grants/db/migrate.js
3. The grants_migrations table tracks which migrations have been applied.
4. Migrations are idempotent — running the command again skips already-applied migrations.
5. New migrations are applied automatically during deployment via the deploy script.

TIP

Always back up the database before running migrations in production. While migrations are designed to be safe, having a backup ensures you can recover from any unexpected issues.

Backup & Recovery

Recommended backup strategy:

1. Schedule daily pg_dump backups: pg_dump -Fc grantcore > backup_$(date +%Y%m%d).dump
2. Store backups in a separate location (off-server or cloud storage).
3. Retain at least 30 days of daily backups.
4. Test recovery periodically: pg_restore -d grantcore_test backup.dump
5. Back up the uploads/ directory separately (user-uploaded files).

CHAPTER 14

Nginx & SSL Configuration

GrantCore.ai uses Nginx as a reverse proxy to handle HTTPS termination and serve static files.

Reverse Proxy Setup

The Nginx configuration routes requests to the backend API and frontend static files:

- /api/* — Proxied to the Node.js backend on port 3009
- /grants/* — Served directly from frontend/public/grants/
- / — Redirects to /grants/login.html

SSL/TLS

HTTPS is strongly recommended for production deployments:

1. Obtain an SSL certificate from Let’s Encrypt using certbot.
2. Configure Nginx to listen on port 443 with SSL.
3. Redirect all HTTP (port 80) traffic to HTTPS.
4. Set up automatic certificate renewal with a cron job.

Service Management

The GrantCore.ai backend runs as a systemd service:

- Start: sudo systemctl start grantcore-ai
- Stop: sudo systemctl stop grantcore-ai
- Restart: sudo systemctl restart grantcore-ai
- Status: sudo systemctl status grantcore-ai
- Logs: sudo journalctl -u grantcore-ai -f

PART

API Reference

CHAPTER 15

Admin API Endpoints

All admin API endpoints require authentication (JWT token) and administrator role. Base URL: /api/admin/

User Management Endpoints

GET /api/admin/users

List all user accounts. Returns array of user objects with id, email, name, role, is_active, created_at, and workspace_id.

POST /api/admin/users

Create a new user account. Required fields: email, password (min 8 chars), name, role (RESEARCHER or ADMIN). Returns the created user object. Returns 409 if email already exists.

PATCH /api/admin/users/:id

Update a user's name, role, or is_active status. Returns the updated user object.

PATCH /api/admin/users/:id/password

Reset a user's password. Required field: password (min 8 chars). Returns success confirmation.

DELETE /api/admin/users/:id

Delete a user account. Cascades to activation_codes and workspace_users. Returns 400 if attempting to delete own account.

POST /api/admin/users/:id/activate

Generate and send an activation code. Required field: expiryDays (number, 0 for permanent). Returns the generated code and expiry information.

Audit Endpoints

GET /api/admin/audit

Get audit log entries. Query parameters: email (ILIKE search), days (time range), success (boolean filter), limit (default 200, max 1000). Returns array of audit entries.

GET /api/admin/audit/stats

Get aggregated audit statistics. Returns: last_24h, last_24h_success, last_24h_failed, last_7d, last_7d_failed, total.

AI Settings Endpoints

GET /api/admin/ai-settings

Get current system AI configuration. Returns ai_provider, ai_model, ai_api_key (masked), and ai_base_url.

PUT /api/admin/ai-settings

Update system AI configuration. Accepts any of: ai_provider, ai_model, ai_api_key, ai_base_url. Updates system_settings table and runtime environment variables. Returns success confirmation.

PART

Troubleshooting

CHAPTER 16

Common Admin Issues

Authentication Issues

Q: Users cannot log in after account creation.
Verify the account is active (status shows "Active" in the users table). If the account is "Pending", either send an activation code or click "Activate Directly".

Q: JWT tokens are not being validated.
Ensure JWT_REQUIRED=true in the backend .env file. When set to false, the server accepts requests without valid tokens, which is only appropriate for development.

Q: Rate limiting is blocking legitimate users.
The default rate limit is 20 login attempts per 15-minute window per IP. If users share a corporate IP (e.g., behind a NAT), this limit may be reached quickly. Adjust the rate limit settings in app.js if needed.

AI Configuration Issues

Q: AI features return "Missing credentials" errors.
Verify the AI provider is configured in Admin: Settings. Click "Test Connection" to verify. If the test fails, check that your API key is valid and has sufficient credits/quota with the provider.

Q: The "Test Connection" button fails but the API key is correct.
Check network connectivity from the server to the AI provider's API. Some firewalls or proxy servers may block outbound HTTPS connections. Verify the provider's API status page for outages.

Q: Users report AI features work intermittently.
Check the provider's rate limits and quota. Claude and OpenAI have per-minute and per-day token limits. Monitor the server logs for specific error messages: journalctl -u grantcore-ai | grep "AI error"

Database Issues

Q: Migrations fail to run.
Check the PostgreSQL connection string in .env. Ensure the database user has CREATE TABLE and ALTER TABLE permissions. Check grants_migrations table for the last successful migration. Review the specific migration SQL file for syntax errors.

Q: "relation does not exist" errors.

Run migrations: node backend/src/grants/db/migrate.js. If migrations have been skipped, check the grants_migrations table and manually mark any pre-existing migrations as applied.

Deployment Issues

Q: The deploy script fails with permission errors.

Ensure SSH key authentication is set up for the deployment user. The deploy user needs write access to /var/www/grantcore-ai/ and sudo access for systemctl and nginx commands.

Q: Changes are not visible after deployment.

Clear browser cache with Ctrl+Shift+R. Static files may be cached by Nginx—check the Nginx cache configuration. Verify the systemd service restarted successfully: systemctl status grantcore-ai.

CHAPTER 17

Monitoring & Maintenance

Server Health Checks

Monitor these indicators regularly:

- Application status: systemctl status grantcore-ai (should show "active (running)")
- Response time: curl -w "%{time_total}" https://your-domain.com/api/auth/me
- Disk space: df -h (ensure /var/www has sufficient space for uploads)
- Memory usage: free -m (Node.js process typically uses 150–300MB)
- Database connections: SELECT count(*) FROM pg_stat_activity WHERE datname='grantcore';

Log Files

- Application logs: journalctl -u grantcore-ai -f (live tail)
- Nginx access logs: /var/log/nginx/access.log
- Nginx error logs: /var/log/nginx/error.log
- PostgreSQL logs: /var/log/postgresql/ (check for slow queries)

Regular Maintenance Tasks

- Weekly: Review audit trail for suspicious activity
- Weekly: Check server disk space and clean old uploads if needed
- Monthly: Review and update AI provider API key usage and billing
- Monthly: Run database VACUUM ANALYZE for performance optimization
- Quarterly: Review user accounts and disable inactive accounts
- Quarterly: Update Node.js and npm packages for security patches
- Annually: Rotate JWT_SECRET and regenerate all tokens
- Annually: Renew SSL certificates (or verify auto-renewal)

PART

Appendix

CHAPTER 18

Quick Reference Card

Default URLs

- Login: /grants/login.html
- Register: /grants/register.html
- Dashboard: /grants/dashboard.html
- Admin Users: /grants/admin_users.html
- Admin Settings: /grants/admin_settings.html
- Admin Funders: /grants/admin_funders.html
- Admin Audit: /grants/admin_audit.html

Admin API Endpoints

- GET /api/admin/users — List users
- POST /api/admin/users — Create user
- PATCH /api/admin/users/:id — Update user
- PATCH /api/admin/users/:id/password — Reset password
- DELETE /api/admin/users/:id — Delete user
- POST /api/admin/users/:id/activate — Send activation code
- GET /api/admin/audit — Get audit log
- GET /api/admin/audit/stats — Get audit stats
- GET /api/admin/ai-settings — Get AI config
- PUT /api/admin/ai-settings — Update AI config

Database Migration Commands

- Run migrations: node backend/src/grants/db/migrate.js
- Check migration status: SELECT * FROM grants_migrations ORDER BY id;
- Backup database: pg_dump -Fc grantcore > backup.dump
- Restore database: pg_restore -d grantcore backup.dump

Service Commands

- Start: sudo systemctl start grantcore-ai
- Stop: sudo systemctl stop grantcore-ai
- Restart: sudo systemctl restart grantcore-ai

- Status: sudo systemctl status grantcore-ai
- Logs: sudo journalctl -u grantcore-ai -f
- Nginx reload: sudo nginx -t && sudo systemctl reload nginx

CHAPTER 19

Internationalization (i18n)

All admin pages support multiple languages through the built-in i18n system.

Supported Languages

- English (EN) — Default language
- French (FR) — Français
- German (DE) — Deutsch
- Spanish (ES) — Español
- Italian (IT) — Italiano
- Portuguese (PT) — Português
- Arabic (AR) — bvDc–1b†Jb’ ‡v—F, TL layout)

Users can switch languages using the language toggle in the top navigation bar. The selection is persisted in localStorage and applied across all pages including admin pages.

Translation Keys

Admin pages use the same i18n system as researcher pages. All labels, buttons, table headers, and messages are translated. The translation file is located at frontend/public/grants/grants_i18n.js.

CHAPTER 20

Contact & Support

GrantCore.ai is developed and maintained by SoftSim Technologies Inc., based in Montreal, Canada.

Technical Support

- Email: info@softsim.ca
- Website: https://grantcore.ai
- Documentation: Included in User Guide and Admin Guide PDFs

Reporting Issues

When reporting issues, please include:

- Browser and version (e.g., Chrome 120, Safari 17)
- Operating system (e.g., Windows 11, macOS Sonoma)
- Steps to reproduce the issue
- Any error messages displayed on screen
- Server logs if applicable (journalctl -u grantcore-ai)

About SoftSim Technologies

SoftSim Technologies Inc. builds AI-powered tools for research, healthcare, and education. Our mission is to make grant funding more accessible and efficient for researchers worldwide by leveraging artificial intelligence throughout the grant lifecycle.