

SheAware Project Documentation

Table of Contents

- 1. Project Overview
- 2. How to Run the Project
- 3. Project Architecture
- 4. Folder Structure
- 5. API Documentation
- 6. Database Schema
- 7. Django Admin

Project Overview

SheAware is a Django REST Framework-based backend application designed to help women track gynecological symptoms, assess health risks using AI, access educational content, manage notifications, and find support resources.

Attribute	Details
Framework	Django 5.1.7
API Framework	Django REST Framework 3.14+
Authentication	JWT (Simple JWT 5.3+)
Database	SQLite (Development)
AI Integration	Google Generative AI (Gemini)
Python Version	Python 3.12+

How to Run the Project

Prerequisites

Requirement	Version
Python	3.12+
pip	Latest
Virtual Environment	Recommended

Installation Commands

Step	Command	Description
1. Clone Repository	<code>git clone <repository-url></code>	Clone the project repository
2. Navigate to Directory	<code>cd she_aware</code>	Enter project directory
3. Create Virtual Environment	<code>python -m venv venv</code>	Create isolated Python environment
4. Activate Virtual Environment (Mac/Linux)	<code>source venv/bin/activate</code>	Activate virtual environment
5. Activate Virtual Environment (Windows)	<code>venv\Scripts\activate</code>	Activate virtual environment
6. Install Dependencies	<code>pip install -r requirements.txt</code>	Install all required packages
7. Run Migrations	<code>python manage.py migrate</code>	Apply database migrations
8. Create Superuser	<code>python manage.py createsuperuser</code>	Create admin account
9. Run Development Server	<code>python manage.py runserver</code>	Start server at <code>http://127.0.0.1:8000</code>

Additional Commands

Command	Purpose
<code>python manage.py makemigrations</code>	Create new database migrations
<code>python manage.py migrate</code>	Apply database migrations
<code>python manage.py createsuperuser</code>	Create admin user
<code>python manage.py runserver</code>	Run development server
<code>python manage.py runserver 0.0.0.0:8000</code>	Run server accessible from network
<code>python manage.py shell</code>	Open Django shell
<code>python manage.py test</code>	Run tests
<code>python manage.py collectstatic</code>	Collect static files for production

Environment Configuration

Variable	Description	Required
SECRET_KEY	Django secret key	Yes
DEBUG	Debug mode (True/False)	Yes
ALLOWED_HOSTS	Comma-separated allowed hosts	Yes
GEMINI_API_KEY	Google Generative AI API key	Yes (for AI features)

Project Architecture

Architecture Flow

Client Layer (Mobile/Web App) - Sends HTTP/HTTPS requests to the backend API

API Layer (Django REST Framework) - Custom JSON Renderer & Exception Handler (apps.common) - Routes requests to appropriate Django applications

Application Layer - Users App: Authentication & device management - Symptoms App: Symptom tracking & risk assessment - Education App: Content management - Notifications App: Notification scheduling - Support App: Resource management

Data Layer - SQLite Database (Development) - Stores: User, Device, SymptomEntry, Article, Notification, Resource data

External Services - Google Generative AI (Gemini) for AI-powered risk scoring

Application Components

Component	Responsibility
config/	Project settings, URL routing, WSGI/ASGI configuration
apps.common/	Shared utilities (custom renderers, exception handlers)
apps.users/	User authentication, device registration
apps.symptoms/	Symptom tracking, AI-powered risk assessment
apps.education/	Educational articles and content management
apps.notifications/	Notification scheduling and push notifications

Component	Responsibility
apps.support/	Support resources (helplines, NHS resources)

Authentication Flow

Step	Action	Endpoint
1	Device registers with platform	POST /v1/auth/device
2	Server returns device_id	Response: {"device_id": "uuid"}
3	Client stores device_id	Local storage
4	Client sends device_id in header	X-Device-Id: <uuid>
5	Server identifies user/device	Middleware/View logic

Folder Structure

Root Directory

File/Folder	Type	Description
apps/	Directory	Contains all Django applications
config/	Directory	Project configuration files
db.sqlite3	File	SQLite database file
manage.py	File	Django management script
requirements.txt	File	Python dependencies
sheaware_postman_collection.json	File	Postman API collection

apps/ Directory Structure

App	Purpose	Key Files
common/	Shared utilities	renderers.py, exceptions.py
users/	User & device management	models.py, views.py, admin.py
symptoms/	Symptom tracking & AI scoring	models.py, views.py, services/scoring.py
education/	Educational content	models.py, views.py, admin.py
notifications/	Notification system	models.py, views.py, admin.py

App	Purpose	Key Files
support/	Support resources	models.py, views.py, serializers.py

config/ Directory

File	Purpose
settings.py	Django settings (database, apps, middleware, JWT config)
urls.py	Root URL configuration
wsgi.py	WSGI application entry point
asgi.py	ASGI application entry point

Detailed Folder Tree

```

she_aware/
  apps/
    __init__.py
    common/
      __init__.py
      exceptions.py      # Custom exception handler
      renderers.py      # Standard JSON renderer
    users/
      __init__.py
      admin.py           # User & Device admin
      apps.py
      models.py          # User, Device models
      serializers.py
      urls.py
      views.py           # Device registration
      migrations/
    symptoms/
      __init__.py
      admin.py           # SymptomEntry, RiskScore admin
      apps.py
      models.py          # SymptomEntry, RiskScore, SymptomDefinition
      serializers.py
      urls.py
      views.py           # Symptom CRUD, scoring endpoints
      services/
        scoring.py      # AI risk scoring service
      migrations/
    education/

```

```

__init__.py
admin.py          # Article admin
apps.py
models.py         # Article model
serializers.py
urls.py
views.py         # Article list/detail views
migrations/
notifications/
__init__.py
admin.py         # Notification admin
apps.py
models.py        # Notification model
urls.py
views.py         # Schedule & push notifications
migrations/
support/
__init__.py
apps.py
models.py        # Resource model
serializers.py
urls.py
views.py         # Resource viewset
migrations/
config/
__init__.py
settings.py      # Project settings
urls.py          # Root URL configuration
wsgi.py
asgi.py
db.sqlite3       # SQLite database
manage.py        # Django CLI
requirements.txt  # Dependencies
sheaware_postman_collection.json # API collection

```

API Documentation

Base URL

Environment	Base URL
Development	http://127.0.0.1:8000/v1
Production	https://your-domain.com/v1

Standard Response Format

All API responses follow this standard format:

```
{
  "success": true,
  "data": { ... },
  "message": "Optional message",
  "errors": null
}
```

API Endpoints

1. Authentication & Device Management

Endpoint	Method	Description	Auth Required	Request Body	Response
/auth/device	POST	Register a new device	No	{"platform": {"device_id": "ios/android/web", "server_ack": true}}	

Example Request:

```
curl -X POST http://127.0.0.1:8000/v1/auth/device \
  -H "Content-Type: application/json" \
  -d '{"platform": "ios"}'
```

Example Response:

```
{
  "success": true,
  "data": {
    "device_id": "550e8400-e29b-41d4-a716-446655440000",
    "server_ack": true
  }
}
```

2. Symptoms Management

Endpoint	Method	Description	Auth Required	Headers	Request Body
/symptoms	POST	Log new symptom entry	No	X-Device-ID: <uuid>	See below
/symptoms	GET	Get symptom history	No	X-Device-ID: <uuid>	N/A
/score/latest	GET	Get latest risk score	No	X-Device-ID: <uuid>	N/A
/score/history	GET	Get risk score history	No	X-Device-ID: <uuid>	N/A

POST /symptoms - Request Body:

```
{
  "device_id": "550e8400-e29b-41d4-a716-446655440000",
  "symptoms": [
    {
      "name": "abnormal_bleeding",
      "severity": 3,
      "duration_value": 5,
      "duration_unit": "days"
    }
  ],
  "pain_intensity": 4,
  "mood": "Sad",
  "notes": "Additional notes"
}
```

Symptom Fields:

Field	Type	Required	Description
device_id	UUID	Yes	Device identifier
symptoms	Array	Yes	List of symptom objects
symptoms[].name	String	Yes	Symptom identifier
symptoms[].severity	Integer (1-5)	Yes	Severity level
symptoms[].duration_value	Integer	Yes	Duration value
symptoms[].duration_unit	String	Yes	Unit: "days", "weeks", "months"
pain_intensity	Integer (1-10)	No	Pain level

Field	Type	Required	Description
mood	String	No	Current mood
notes	Text	No	Additional notes

GET /score/latest - Response:

```
{
  "success": true,
  "data": {
    "id": 1,
    "score": 65,
    "category": "Moderate",
    "flags": ["abnormal_bleeding", "prolonged_duration"],
    "explanation": "Your symptoms suggest moderate risk...",
    "model": "gemini-v1",
    "created_at": "2024-01-20T10:30:00Z"
  }
}
```

Risk Score Categories:

Score Range	Category	Description
0-25	Low	Minimal concern
26-50	Moderate	Monitor symptoms
51-75	Elevated	Consider consultation
76-100	High	Seek medical attention

3. Education & Content

Endpoint	Method	Description	Auth Required	Query Parameters
/content/articles	GET	List all published articles	No	category, lang
/content/articles/<id>	GET	Get article details	No	N/A

Query Parameters:

Parameter	Type	Description	Example
category	String	Filter by category	?category=Symptoms
lang	String	Filter by language	?lang=en

GET /content/articles - Response:

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "title": "Understanding PCOS",
      "slug": "understanding-pcos",
      "category": "Symptoms",
      "body_html": "<p>Article content...</p>",
      "sources": ["https://nhs.uk/..."],
      "tags": ["pcos", "hormones"],
      "lang": "en",
      "published_at": "2024-01-15T00:00:00Z"
    }
  ]
}
```

Article Categories:

Category	Description
General	General health information
Symptoms	Symptom-specific articles
Treatment	Treatment options
Prevention	Preventive care
Lifestyle	Lifestyle recommendations

4. Notifications

Endpoint	Method	Description	Auth Required	Request Body
/notifications	POST	Schedule a reminder	No	See below
/notifications	POST	Send push notification (Admin)	Yes (Admin)	N/A

POST /notifications/schedule - Request Body:

```
{
  "device_id": "550e8400-e29b-41d4-a716-446655440000",
  "type": "reminder",
  "title": "Log check",
  "message": "Time to check in",
  "cron_spec": "0 9 * * *"
}
```

Notification Fields:

Field	Type	Required	Description
device_id	UUID	Yes	Device identifier
type	String	Yes	“reminder”, “campaign”, “system”
title	String	Yes	Notification title
message	Text	Yes	Notification message
cron_spec	String	No	Cron format for recurring reminders

Cron Format Examples:

Cron Expression	Description
0 9 * * *	Daily at 9:00 AM
0 9 * * 1	Every Monday at 9:00 AM
0 9 1 * *	First day of month at 9:00 AM
0 */6 * * *	Every 6 hours

5. Support Resources

Endpoint	Method	Description	Auth Required	Query Parameters
/resources/	GET	List all active resources	No	N/A
/resources/<id>	GET	Get resource details	No	N/A

GET /resources/ - Response:

```
{
  "success": true,
  "data": [
```

```

{
  "id": 1,
  "title": "NHS Direct",
  "description": "24/7 health advice",
  "category": "NHS",
  "link": "https://nhs.uk",
  "phone_number": "111",
  "is_active": true,
  "created_at": "2024-01-01T00:00:00Z"
}
]
}

```

Resource Categories:

Category	Description
NHS	NHS & Medical Resources
HELPLINE	Helplines & Support
EMERGENCY	Emergency Contact
OTHER	Other Resources

API Error Responses

Status Code	Error Type	Example Response
400	Bad Request	{"success": false, "errors": {"field": ["Error message"]}}
401	Unauthorized	{"success": false, "message": "Authentication required"}
403	Forbidden	{"success": false, "message": "Permission denied"}
404	Not Found	{"success": false, "message": "Resource not found"}
500	Server Error	{"success": false, "message": "Internal server error"}

Database Schema

Database Relationships Overview

User and Device Relationships: - Users can have multiple SymptomEntries (One-to-Many) - Devices can have multiple SymptomEntries (One-to-Many) - Users can have multiple Notifications (One-to-Many) - Devices can have multiple Notifications (One-to-Many)

Symptom Tracking: - Each SymptomEntry has exactly one RiskScore (One-to-One) - RiskScore is automatically created when a SymptomEntry is saved

Content and Resources: - Articles are standalone (no foreign key relationships) - Resources are standalone (no foreign key relationships)

Database Tables

1. users__user

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	User ID
email	VARCHAR(254)	UNIQUE, NOT NULL	User email (username)
password	VARCHAR(128)	NOT NULL	Hashed password
first_name	VARCHAR(150)	NULL	First name
last_name	VARCHAR(150)	NULL	Last name
is_staff	Boolean	DEFAULT FALSE	Staff status
is_active	Boolean	DEFAULT TRUE	Active status
is_superuser	Boolean	DEFAULT FALSE	Superuser status
date_joined	DateTime	AUTO NOW ADD	Registration date
deleted_at	DateTime	NULL	Soft delete timestamp

Indexes: - PRIMARY KEY on id - UNIQUE INDEX on email

2. users__device

Column	Type	Constraints	Description
id	UUID	PRIMARY KEY	Device UUID
platform	VARCHAR(20)	NOT NULL	Platform: android/ios/web
created_at	DateTime	AUTO NOW ADD	Registration timestamp
last_seen	DateTime	AUTO NOW	Last activity timestamp

Indexes: - PRIMARY KEY on id

Choices: | Value | Label | |———|———| | android | Android | | ios | iOS
| | web | Web |

3. symptoms_symptomdefinition

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	Definition ID
key	VARCHAR(50)	UNIQUE, NOT NULL	Internal key (e.g., 'abnormal_bleeding')
display_text	VARCHAR(100)	NOT NULL	Display name
created_at	DateTime	AUTO NOW ADD	Creation timestamp

Indexes: - PRIMARY KEY on id - UNIQUE INDEX on key

4. symptoms_symptumentry

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	Entry ID
device_id	UUID	FOREIGN KEY (users_device), NULL	Device reference
user_id	BigInteger	FOREIGN KEY (users_user), NULL	User reference
entry_time	DateTime	DEFAULT NOW	Symptom entry time
symptoms	JSON	DEFAULT []	Symptom data array
pain_intensity	Integer	NULL	Pain level (1-10)
mood	VARCHAR(50)	NULL	Mood description
notes	Text	NULL	Additional notes
created_at	DateTime	AUTO NOW ADD	Creation timestamp
deleted_at	DateTime	NULL	Soft delete timestamp

Indexes: - PRIMARY KEY on id - INDEX on (device_id, entry_time) - INDEX on (user_id, entry_time)

Foreign Keys: - device_id → users_device.id (SET NULL on delete) - user_id → users_user.id (SET NULL on delete)

JSON Structure (symptoms field):

```
[
  {
    "name": "abnormal_bleeding",
    "severity": 3,
    "duration_value": 5,
    "duration_unit": "days"
  }
]
```

5. symptoms_riskscore

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	Score ID
entry_id	BigInteger	FOREIGN KEY (symp-toms_symptumentry), UNIQUE	Symptom entry reference
score	Integer	NOT NULL	Risk score (0-100)
category	VARCHAR(50)	NOT NULL	Risk category
flags	JSON	DEFAULT []	Risk flags array
explanation	Text	NULL	AI-generated explanation
model	VARCHAR(50)	DEFAULT 'gemini-v1'	AI model used
response_payload	JSON	NULL	Raw AI response
created_at	DateTime	AUTO NOW ADD	Creation timestamp

Indexes: - PRIMARY KEY on id - UNIQUE INDEX on entry_id

Foreign Keys: - entry_id → symptoms_symptumentry.id (CASCADE on delete)

Category Values: | Category | Score Range | |
 Low | 0-25 | | Moderate | 26-50 | | Elevated | 51-75 | | High | 76-100 |

6. education_article

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	Article ID

Column	Type	Constraints	Description
title	VARCHAR(200)	NOT NULL	Article title
slug	VARCHAR(50)	UNIQUE, NOT NULL	URL-friendly slug
category	VARCHAR(50)	DEFAULT 'General'	Article category
body_html	Text	NOT NULL	HTML content
sources	JSON	DEFAULT []	Source URLs
tags	JSON	DEFAULT []	Tags array
lang	VARCHAR(10)	DEFAULT 'en'	Language code
is_published	Boolean	DEFAULT FALSE	Publication status
published_at	DateTime	NULL	Publication timestamp
updated_at	DateTime	AUTO NOW	Last update timestamp
created_at	DateTime	AUTO NOW ADD	Creation timestamp

Indexes: - PRIMARY KEY on `id` - UNIQUE INDEX on `slug` - INDEX on `is_published` - INDEX on `category` - INDEX on `lang`

7. notifications_notification

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	Notification ID
device_id	UUID	FOREIGN KEY (users_device), NULL	Device reference
user_id	BigInteger	FOREIGN KEY (users_user), NULL	User reference
type	VARCHAR(20)	DEFAULT 'reminder'	Notification type
title	VARCHAR(100)	NOT NULL	Notification title
message	Text	NOT NULL	Notification message
cron_spec	VARCHAR(50)	NULL	Cron schedule
next_send	DateTime	NULL	Next send time
is_sent	Boolean	DEFAULT FALSE	Sent status
payload	JSON	DEFAULT {}	Additional data
created_at	DateTime	AUTO NOW ADD	Creation timestamp

Indexes: - PRIMARY KEY on `id` - INDEX on `type` - INDEX on `is_sent`

Foreign Keys: - `device_id` → `users_device.id` (CASCADE on delete) - `user_id` → `users_user.id` (CASCADE on delete)

Type Choices: | **Value** | **Label** | |————|————| | reminder | Reminder | |
campaign | Campaign | | system | System |

8. support_resource

Column	Type	Constraints	Description
id	BigInteger	PRIMARY KEY, AUTO INCREMENT	Resource ID
title	VARCHAR(255)	NOT NULL	Resource title
description	Text	NULL	Resource description
category	VARCHAR(50)	DEFAULT 'OTHER'	Resource category
link	VARCHAR(200)	NULL	Resource URL
phone_number	VARCHAR(50)	NULL	Contact number
is_active	Boolean	DEFAULT TRUE	Active status
created_at	DateTime	AUTO NOW ADD	Creation timestamp
updated_at	DateTime	AUTO NOW	Last update timestamp

Indexes: - PRIMARY KEY on id - INDEX on category - INDEX on is_active

Category Choices: | **Value** | **Label** | |————|————| | NHS | NHS &
Medical Resources | | HELPLINE | Helplines & Support | | EMERGENCY |
Emergency Contact | | OTHER | Other Resources |

Database Relationships Summary

Relationship	From Table	To Table	Type	On Delete
User → SymptomEntry	users_user	symptoms_symptomentry	One-to-Many	SET NULL
Device → SymptomEntry	users_device	symptoms_symptomentry	One-to-Many	SET NULL
SymptomEntry → RiskScore	symptoms_symptomentry	symptoms_riskscore	One-to-One	CASCADE
Device → Notification	users_device	notifications_notification	One-to-Many	CASCADE
User → Notification	users_user	notifications_notification	One-to-Many	CASCADE

Django Admin

Admin Panel Access

Attribute	Details
URL	http://127.0.0.1:8000/admin/
Login	Use superuser credentials
Create Superuser	python manage.py createsuperuser

Registered Models

1. Users App

Model	Admin Class	List Display	Features
User	Default	email, first_name, last_name, is_staff, is_active	User management, password reset
Device	Default	id, platform, created_at, last_seen	Device tracking

User Admin Features: - Create/edit users - Change passwords - Manage permissions and groups - Filter by staff status, active status - Search by email, first name, last name

2. Symptoms App

Model	Admin Class	List Display	Features
SymptomEntry	SymptomEntryAdmin	id, entry_time, user, device	Inline RiskScore display
RiskScore	Default	id, score, category, created_at	View AI-generated scores
SymptomDefinition	Default	key, display_text	Manage symptom types

SymptomEntry Admin Features: - View symptom entries with associated risk scores - Inline editing of risk scores - Filter by user, device, entry date - Search by notes - JSON field display for symptoms

RiskScore Inline: - Displays risk score directly in symptom entry admin - Shows score, category, flags, explanation - View raw AI response payload

3. Education App

Model	Admin Class	List Display	Features
Article	ArticleAdmin	title, category, lang, is_published, updated_at	Rich content management

Article Admin Features: - Create/edit articles with HTML content - Filter by: - Category - Language - Publication status - Search by: - Title - Body content - Bulk actions: - Publish/unpublish articles - Auto-populate slug from title - JSON field editing for sources and tags

4. Notifications App

Model	Admin Class	List Display	Features
Notification	NotificationAdmin	title, type, device, next_send, is_sent	Notification management

Notification Admin Features: - Create/schedule notifications - Filter by: - Type (reminder, campaign, system) - Sent status - View scheduled notifications - Edit cron specifications - JSON payload editing

5. Support App

Model	Admin Class	List Display	Features
Resource	Default	title, category, link, phone_number, is_active	Resource management

Resource Admin Features: - Add/edit support resources - Filter by category and active status - Manage NHS resources, helplines, emergency contacts - Toggle active status for visibility control

Admin Customizations

Custom JSON Renderer All API responses use `StandardJSONRenderer` from `apps.common.renderers`: - Wraps responses in standard format - Adds `success`, `data`, `message`, `errors` fields - Consistent error handling

Custom Exception Handler `custom_exception_handler` from `apps.common.exceptions`:
- Formats all exceptions consistently - Returns standardized error responses -
Handles validation errors, authentication errors, etc.

Inline Admin Classes

Inline Class	Parent Model	Inline Model	Type
RiskScoreInline	SymptomEntry	RiskScore	StackedInline

Admin Workflow Examples

Creating Educational Content

1. Navigate to **Education** → **Articles**
2. Click **Add Article**
3. Fill in:
 - Title
 - Slug (auto-generated)
 - Category
 - Body HTML
 - Sources (JSON array)
 - Tags (JSON array)
 - Language
4. Check **Is Published** when ready
5. Save

Managing Symptom Definitions

1. Navigate to **Symptoms** → **Symptom Definitions**
2. Click **Add Symptom Definition**
3. Enter:
 - Key (e.g., `abnormal_bleeding`)
 - Display Text (e.g., “Abnormal Bleeding”)
4. Save

Viewing Risk Scores

1. Navigate to **Symptoms** → **Symptom Entries**

2. Click on an entry
3. View inline **Risk Score** section showing:
 - Score value
 - Category
 - Flags
 - AI explanation
 - Model used
 - Raw response payload

Scheduling Notifications

1. Navigate to **Notifications** → **Notifications**
2. Click **Add Notification**
3. Fill in:
 - Device/User
 - Type
 - Title & Message
 - Cron spec (for recurring)
4. Save

Admin Permissions

Permission Level	Can Access	Can Modify
Superuser	All models	All models
Staff (with permissions)	Assigned models	Assigned models
Regular User	No admin access	N/A

Additional Notes

Security Considerations

Feature	Implementation
Authentication	JWT tokens (1 hour access, 7 day refresh)
Password Hashing	Django's default PBKDF2
CSRF Protection	Enabled for web requests
SQL Injection	Protected via Django ORM
XSS Protection	Django template escaping

API Rate Limiting

Currently not implemented. Consider adding for production: - Django REST Framework throttling - Redis-based rate limiting

Monitoring & Logging

- Django's default logging
- Consider adding: Sentry, CloudWatch, or similar

Deployment Checklist

Task	Command/Action
Set DEBUG=False	Update <code>settings.py</code>
Configure ALLOWED_HOSTS	Add production domains
Use PostgreSQL/MySQL	Update DATABASES in settings
Collect static files	<code>python manage.py collectstatic</code>
Set up HTTPS	Configure SSL certificates
Configure CORS	Install django-cors-headers
Set environment variables	Use .env file or system env vars
Run migrations	<code>python manage.py migrate</code>
Create superuser	<code>python manage.py createsuperuser</code>