Photos.stsgym.com - Architecture Review & Feature Documentation

Date: March 22, 2026

Table of Contents

Overview

photos.stsgym.com is a professional photo gallery platform designed for photographers and event organizers. It allows users to upload, organize, and share photos with clients through customizable galleries.

| Property | Value |

|----------|-------|

| **Primary URL** | https://photos.stsgym.com |

| **Auth URL** | https://auth.stsgym.com |

| **Technology Stack** | Node.js Express + Python Flask |

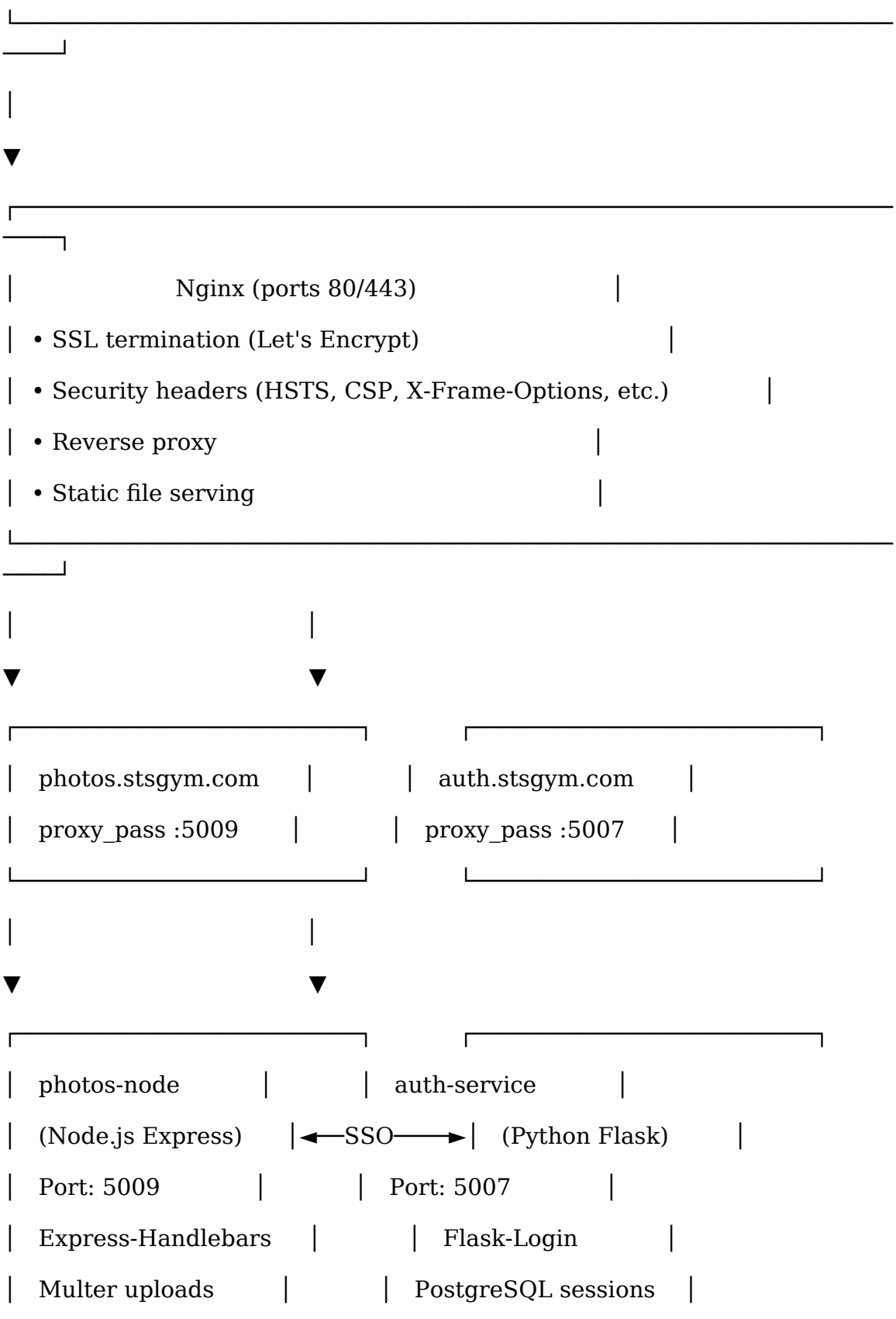
| **Database** | PostgreSQL 15 |

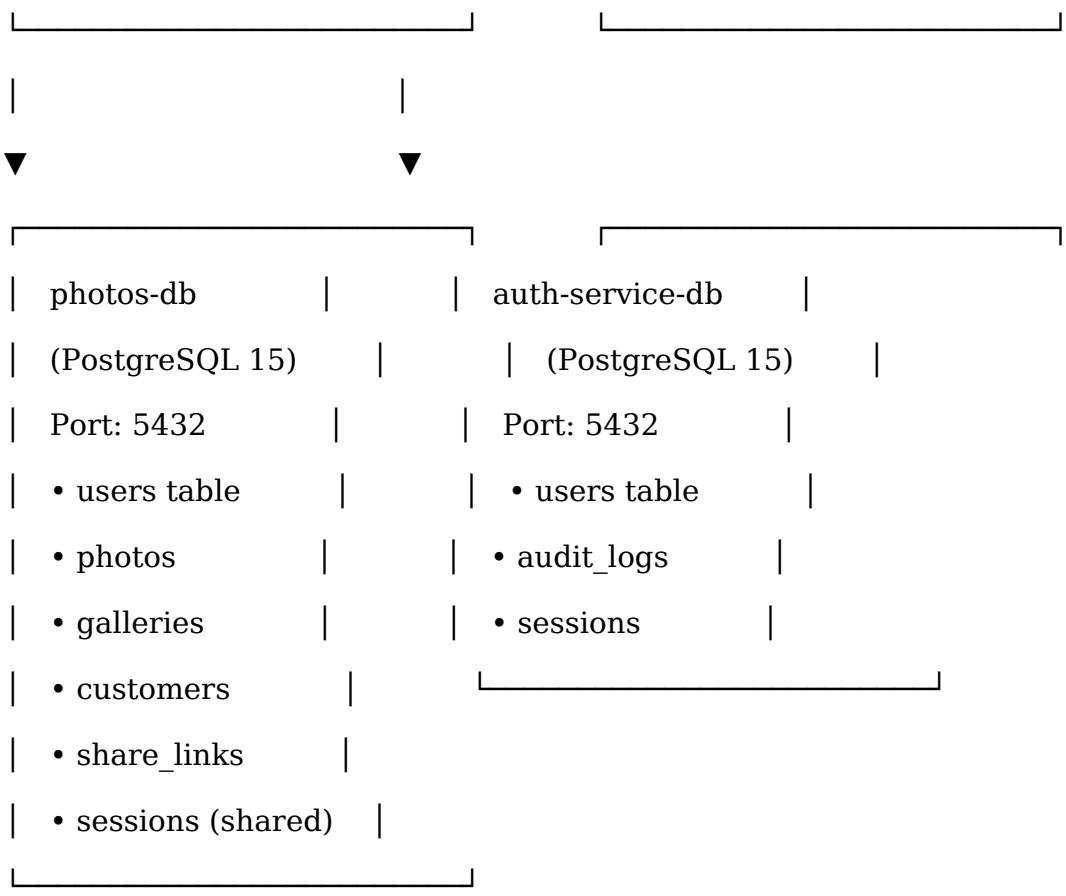
| **Session Store** | PostgreSQL (shared) |

| **File Storage** | Local filesystem (`/uploads/`) |

Architecture

```

┌──────────────────────────────────────────────────────────────────────────────┐

│                          Cloudflare DNS                                      │

│     photos.stsgym.com → 207.244.226.151                                      │

│     auth.stsgym.com → 207.244.226.151                                        │
```


Nginx (ports 80/443)
• SSL termination (Let's Encrypt)
• Security headers (HSTS, CSP, X-Frame-Options, etc.)
• Reverse proxy
• Static file serving
photos.stsgym.com
proxy_pass :5009
auth.stsgym.com
proxy_pass :5007
photos-node
(Node.js Express)
Port: 5009
Express-Handlebars
Multer uploads
SSO
auth-service
(Python Flask)
Port: 5007
Flask-Login
PostgreSQL sessions




```

```

Network: auth-service_default

├── photos-node (connected)

├── auth-service

├── auth-service-db

└── photos-db

Network: photos-node_photos-network

├── photos-node

├── photos-db

└── photos-redis

```
```

| Container | Image | Port | Purpose |

|-----------|-------|------|---------|

| `photos-node` | node:20-alpine | 5009 | Main web application |

| `auth-service` | python:3.11 | 5007 | SSO authentication |

| `photos-db` | postgres:15 | 5432 | Photos database |

| `auth-service-db` | postgres:15 | 5432 | Auth database |

| `photos-redis` | redis:7 | 6379 | Session cache |

Authentication System

The platform uses a **centralized authentication service** (auth.stsgym.com) that provides Single Sign-On (SSO) across all stsgym.com subdomains.

Session Cookie: `.stsgym.com` domain cookie shared between services

Flow:

```

10. User visits photos.stsgym.com
11. photos-node checks for session cookie
12. If no session → redirect to auth.stsgym.com/login
13. User authenticates at auth.stsgym.com
14. auth-service creates session in PostgreSQL
15. User redirected back to photos.stsgym.com with session cookie
16. photos-node validates session via shared database or API
17. User authenticated

```

| Role | Permissions |

|------|-------------|

| `admin` | Full access: manage users, view all galleries, system settings |

| `photographer` | Upload photos, create galleries, manage customers, share galleries |

| `user` | View shared galleries, download photos (with permission) |

| `viewer` | Read-only access to public galleries |

```

18. User → POST /api/forgot-password with email
19. auth-service generates reset token (32 bytes, URL-safe)
20. Token stored in users.reset_token (expires in 1 hour)
21. Email sent via Postfix (172.17.0.1:25)
22. User receives link: /auth/reset-password?token=XXX
23. User → POST /api/reset-password with token + new password
24. Password updated, token cleared, account unlocked

```

Database Schema

```sql

CREATE TABLE users (

id SERIAL PRIMARY KEY,

username VARCHAR(255) UNIQUE NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

password_hash VARCHAR(255) NOT NULL,

user_type VARCHAR(50) DEFAULT 'viewer',

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

last_login TIMESTAMP,

is_active BOOLEAN DEFAULT TRUE,

email_verified BOOLEAN DEFAULT TRUE

);

```

```sql

CREATE TABLE gallery (

id SERIAL PRIMARY KEY,

user_id INTEGER REFERENCES users(id),

name VARCHAR(256) NOT NULL,
```

description TEXT,

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

is_public BOOLEAN DEFAULT FALSE,

cover_photo_id INTEGER REFERENCES photo(id)

);

```

```sql

CREATE TABLE photo (

id SERIAL PRIMARY KEY,

photographer_id INTEGER REFERENCES users(id),

filename VARCHAR(256),

original_path VARCHAR(512),

thumbnail_path VARCHAR(512),

preview_path VARCHAR(512),

title VARCHAR(256),

description TEXT,

upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

rating DOUBLE PRECISION DEFAULT 0.0,

status VARCHAR(20) DEFAULT 'pending',

views INTEGER DEFAULT 0,

archived BOOLEAN DEFAULT FALSE,

locked BOOLEAN DEFAULT FALSE,

download_count INTEGER DEFAULT 0,

gallery_id INTEGER REFERENCES gallery(id)

);

```

```sql
CREATE TABLE customer (
id SERIAL PRIMARY KEY,
photographer_id INTEGER NOT NULL REFERENCES users(id),
email VARCHAR(120) NOT NULL,
name VARCHAR(120),
status VARCHAR(20) DEFAULT 'active',
notes TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
phone VARCHAR(20),
last_accessed TIMESTAMP,
UNIQUE(photographer_id, email)
);
```

```sql
CREATE TABLE gallery_customers (
gallery_id INTEGER REFERENCES gallery(id) ON DELETE CASCADE,
customer_id INTEGER REFERENCES customer(id) ON DELETE CASCADE,
added_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (gallery_id, customer_id)
);
```

```sql
CREATE TABLE share_links (
id SERIAL PRIMARY KEY,
```
```

gallery_id INTEGER REFERENCES gallery(id) ON DELETE CASCADE,

token VARCHAR(64) UNIQUE NOT NULL,

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

expires_at TIMESTAMP,

max_views INTEGER,

views INTEGER DEFAULT 0,

is_active BOOLEAN DEFAULT TRUE,

customer_id INTEGER REFERENCES customer(id),

access_level VARCHAR(20) DEFAULT 'preview', -- 'preview' or 'full'

sender_id INTEGER REFERENCES users(id),

recipient_email VARCHAR(255)

);

```

```sql

CREATE TABLE gallery_permissions (

id SERIAL PRIMARY KEY,

gallery_id INTEGER REFERENCES gallery(id) ON DELETE CASCADE,

user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,

permission_level VARCHAR(20) DEFAULT 'view',  -- 'view', 'edit', 'copy'

granted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

granted_by INTEGER REFERENCES users(id),

UNIQUE(gallery_id, user_id)

);

```

```sql

CREATE TABLE sessions (

sid VARCHAR(255) PRIMARY KEY,

sess JSONB NOT NULL,

expire TIMESTAMP NOT NULL

);

```

Features

- **Technology:** Multer middleware
- **Max Files:** 50 photos per upload
- **Max Size:** 50MB per photo
- **Storage Path:** `/uploads/<user_id>/<filename>`
- **File Naming:** `<timestamp>-<random>.<extension>`

Features:

- Drag-and-drop upload
- Multiple file selection
- Automatic thumbnail generation (planned)
- Preview image creation (planned)
- EXIF metadata extraction (planned)

| Feature | Status | Description |

|---------|--------|-------------|

| Create Gallery | □ | Name, description, public/private |

| Edit Gallery | □ | Update name, description, visibility |

| Delete Gallery | □ | Remove gallery and all photos |

| Set Cover Photo | □ | Choose thumbnail for gallery |

| Add Photos | □ | Upload photos to gallery |

| Remove Photos | □ | Remove individual photos |

| Feature | Status | Description |

|---------|--------|-------------|

| Add Customer | □ | Name, email, phone, notes |

| List Customers | □ | Paginated list |
```

| Delete Customer | □ | Remove from database |

| Auto-Create | □ | Auto-create customer from email when sharing |

| Customer Assignment | □ | Assign customers to galleries |

| Feature | Status | Description |

|---------|--------|-------------|

| Generate Share Link | □ | Unique token for public access |

| Access Levels | □ | Preview (watermarked) or Full (download) |

| Link Expiry | □ | 1/3/7/14/30/90 days or never |

| View Tracking | □ | Count views per link |

| Customer Association | □ | Link share to specific customer |

| Share History | □ | Track all shares with sender/recipient |

| Feature | Status | Description |

|---------|--------|-------------|

| Grant Permission | □ | View/Edit/Copy levels |

| Remove Permission | □ | Revoke access |

| Photographer Access | □ | Grant photographers access to galleries |

Dashboard Cards:

- Total Photos
- Total Customers
- Total Views
- Storage Used

Charts:

- Views over time (line chart)
- Storage usage (doughnut chart)
- Recent activity feed
- Recent galleries

| Feature | Status | Description |

|---------|--------|-------------|

| Dark Mode | □ | Bootstrap dark theme with custom CSS |

| Responsive Design | □ | Mobile-friendly |

| Bootstrap Icons | □ | CSP fixed for CDN fonts |

| Theme Switcher | □ | Light/Dark/Auto modes |

API Reference

| Method | Endpoint | Description |

|--------|----------|-------------|

| GET | `/auth/login` | Redirect to auth.stsgym.com |

| GET | `/auth/logout` | Clear session, redirect |

| GET | `/auth/register` | Redirect to registration |

| GET | `/auth/callback` | Handle SSO callback |

| GET | `/auth/me` | Get current user |

| GET | `/auth/session` | Check session status |

| Method | Endpoint | Description |

|--------|----------|-------------|

| POST | `/api/login` | API login (programmatic) |

| POST | `/api/forgot-password` | Request password reset |

| POST | `/api/reset-password` | Reset password with token |

| GET | `/api/verify-session` | Validate session token |

| Method | Endpoint | Description |

|--------|----------|-------------|

| GET | `/api/stats` | Get dashboard stats |

| POST | `/api/upload` | Upload photos |

| GET | `/api/galleries` | List galleries |

| POST | `/api/galleries` | Create gallery |

| GET | `/api/galleries/:id` | Get gallery details |

| PUT | `/api/galleries/:id` | Update gallery |

| DELETE | `/api/galleries/:id` | Delete gallery |

| Method | Endpoint | Description |

|--------|----------|-------------|

| GET | `/admin` | Admin dashboard |

| GET | `/admin/users` | User management |

| POST | `/admin/user/:id/role` | Change user role |

| GET | `/admin/settings` | System settings |

| Method | Endpoint | Description |

|--------|----------|-------------|

| GET | `/view/:token` | Public gallery view (no auth) |

| GET | `/gallery` | Gallery list page |

| GET | `/` | Home page |

| GET | `/about` | About page |

| GET | `/pricing` | Pricing page |

| GET | `/features` | Features page |

Security

```nginx

Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

X-Frame-Options: SAMEORIGIN

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

Referrer-Policy: strict-origin-when-cross-origin

Permissions-Policy: geolocation=(), microphone=(), camera=()

Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval' cdn.jsdelivr.net; style-src 'self' 'unsafe-inline' cdn.jsdelivr.net fonts.googleapis.com; img-src 'self' data: blob: *.stsgym.com; font-src 'self'
```

fonts.googleapis.com fonts.gstatic.com cdn.jsdelivr.net; connect-src 'self' *.stsgym.com

```

```python

@api_bp.route('/login', methods=['POST'])

@limiter.limit("10 per minute")  # Login attempts

@api_bp.route('/forgot-password', methods=['POST'])

@limiter.limit("5 per minute")  # Password reset requests

```

| Setting | Value |

|---------|-------|

| Cookie Name | `photos_session` |

| Domain | `.stsgym.com` |

| Secure | `true` (production) |

| HttpOnly | `true` |

| SameSite | `lax` |

| Max Age | 7 days |

| Store | PostgreSQL (connect-pg-simple) |

- CSRF tokens generated per session
- Tokens validated on POST/PUT/DELETE requests
- Token exposed to views via `res.locals.csrfToken`

Deployment

- Docker & Docker Compose
- PostgreSQL 15+
- Nginx
- Let's Encrypt SSL certificates

```bash

ssh wez@207.244.226.151

cd /home/wez

git clone https://idm.wezzel.com/crab-meat-repos/photos-stsgym.git photos-node

git clone https://idm.wezzel.com/crab-meat-repos/auth-stsgym.git auth-service

```

photos-node (.env):

```env

NODE_ENV=production

DATABASE_URL=postgresql://photos:photos123@photos-db:5432/photos

SESSION_SECRET=stsgym-photos-session-2026

AUTH_URL=http://auth-service:5005

APP_URL=https://photos.stsgym.com

```

auth-service (.env):

```env

DATABASE_URL=postgresql://auth:auth123@db:5432/auth

SECRET_KEY=[SECRET_KEY_REDACTED]

```

```bash

cd /home/wez/photos-node

docker build -t photos-node_web .

docker compose up -d

cd /home/wez/auth-service

docker compose build --no-cache

docker compose up -d

```
```

```bash
cd /home/wez/photos-node
git pull origin main
docker compose down
docker compose up -d --build
docker logs photos-node --tail 100 -f
```

```bash
sudo certbot --nginx -d photos.stsgym.com
sudo certbot --nginx -d auth.stsgym.com
sudo certbot renew --nginx
```

Troubleshooting

```bash
docker ps | grep -E 'photos|auth'
docker logs photos-node --tail 50
docker logs auth-service --tail 50
netstat -tlnp | grep -E '5009|5007'
```

```bash
grep -A5 'session' /home/wez/photos-node/src/app.js
docker exec photos-db psql -U photos -d photos -c "SELECT * FROM sessions LIMIT 5;"
docker exec photos-node ping -c 1 auth-service
```

```bash
```

ls -la /home/wez/photos-node/uploads/

chmod 755 /home/wez/photos-node/uploads

docker exec photos-db psql -U photos -d photos -c "SELECT * FROM photo LIMIT 5;"

```

```bash

grep '/uploads' /etc/nginx/sites-available/photos.stsgym.com

sudo nginx -t && sudo nginx -s reload

```

```bash

docker ps | grep auth-service

docker exec photos-node wget -q -O - http://auth-service:5005/api/health

docker exec photos-node env | grep AUTH_URL

```

```bash

curl -s https://photos.stsgym.com/health

curl -s https://auth.stsgym.com/api/health

docker exec photos-db psql -U photos -c "SELECT 1"

docker exec auth-service-db psql -U auth -c "SELECT 1"

```

Git Commits (2026-03-19)

| Commit | Description |

|--------|-------------|

| `9889ac4` | Fix session persistence - await session.save() callback |

| `1034f54` | Fix session persistence - enable resave and saveUninitialized |

| `252623e` | Add photo upload functionality with multer |
```

| `4a4c56f` | Add multer dependency for file uploads |

| `bcaef0f` | Add database integration for uploads |

| `a806d2e` | Add customer management - add, list, delete customers |

| `65b4d4b` | Add gallery management - create, view, delete galleries |

| `b67cc8b` | Add proper edit gallery modal |

| `d2f9c8d` | Add customer assignment and public share links |

| `3ea2172` | Add comprehensive gallery sharing system |

Credentials

| Service | Username | Password | Role |

|---------|----------|----------|------|

| photos-node DB | photos | photos123 | - |

| auth-service DB | auth | auth123 | - |

| wez | wez | StsPhotos2026! | admin |

| wezzels | wezzels | newpass2026 | user |

| jrphotos | jrphotos | Kuz-m2if | photographer |

Future Enhancements

25.**Photo Processing Pipeline**

- Automatic thumbnail generation
- Watermark overlay for preview mode
- EXIF metadata extraction

26.**Storage Optimization**

- S3/MinIO integration for scalable storage
- CDN for faster image delivery
- Image compression on upload

27.**Analytics Dashboard**

- Real-time view tracking
- Download statistics

- Customer engagement metrics

28.**Notification System**

- Email notifications for new shares
- SMS alerts (optional)
- In-app notifications

29.**Payment Integration**

- Stripe integration for paid galleries
- Subscription tiers
- Invoice generation

References

- **GitLab Repositories:**
- photos-stsgym: https://idm.wezzel.com/crab-meat-repos/photos-stsgym
- auth-stsgym: https://idm.wezzel.com/crab-meat-repos/auth-stsgym
- photos-flask (archived): https://idm.wezzel.com/crab-meat-repos/photos-flask
- **Related Documentation:**
- `/home/wez/stsgym-work/docs/PHOTOS-STSGYM-COMPLETE-2026-03-19.md`
- `/home/wez/.openclaw/workspace/photos-stsgym/SETUP-GUIDE.md`

Last Updated: 2026-03-19 19:40 UTC