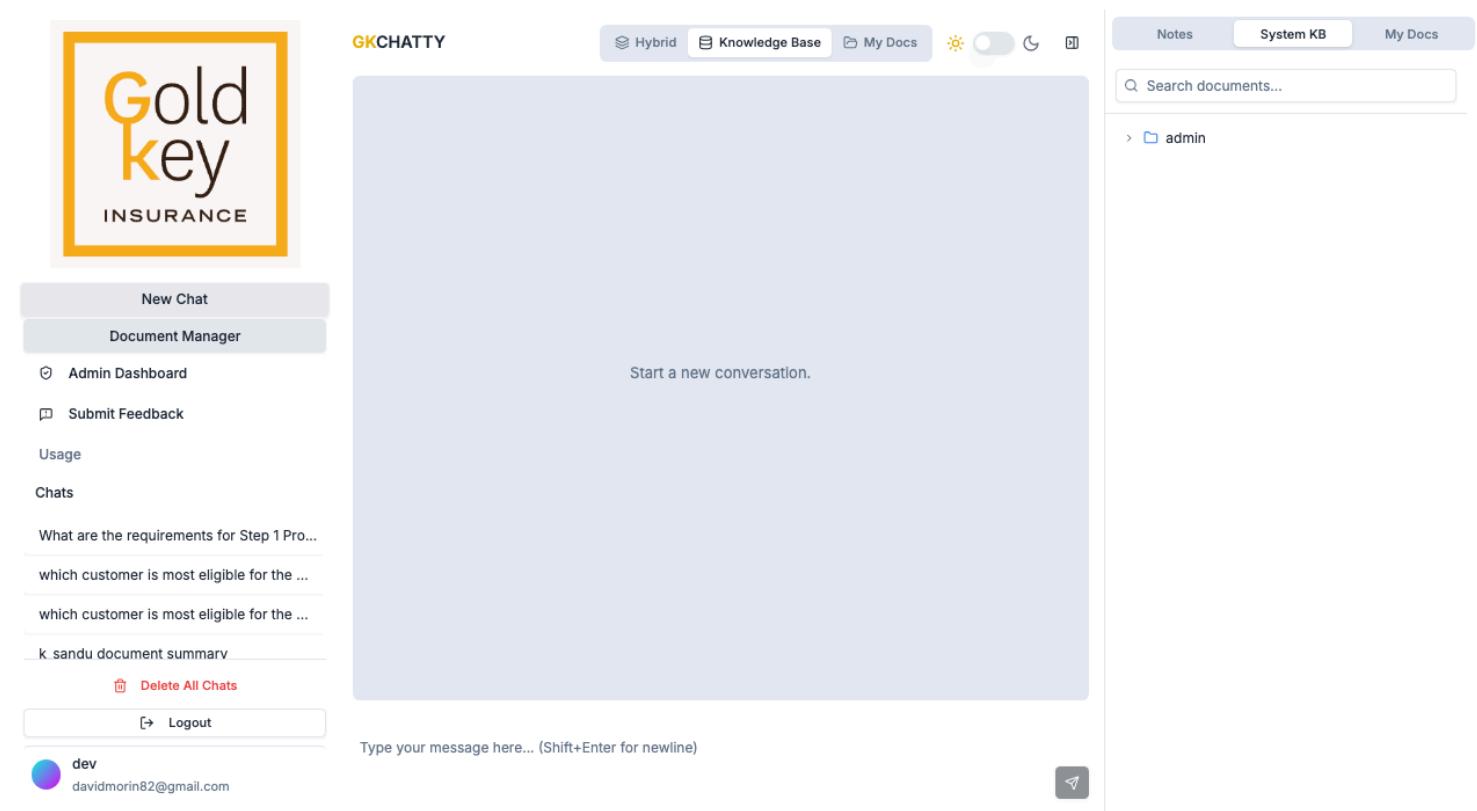


GKChatty Tech Operations Quick Reference

Frontend Interface

GK CHATTY



Architecture

Frontend (Next.js:4003) → Backend (Express:4001) → MongoDB + Pinecone + S3 + OpenAI

Component	Tech	Port/Host
Frontend	Next.js 14, React 18, Tailwind	4003 / Netlify
Backend	Express, TypeScript	4001 / Render
Database	MongoDB (Mongoose)	27017 / Atlas
Vectors	Pinecone	Cloud
Storage	AWS S3 or Local	-
LLM	OpenAI API	Cloud
Cache	Redis (optional)	6379

Environment Variables (Backend .env)

```

# Core
NODE_ENV=production|staging|development
PORT=4001
FRONTEND_URL=https://your-frontend.com

# Database
MONGO_URI=mongodb+srv://user:pass@cluster.mongodb.net/gkchatty

# Auth
JWT_SECRET=<256-bit-hex-key> # crypto.randomBytes(32).toString('hex')

# OpenAI
OPENAI_API_KEY=sk-proj-xxx
OPENAI_PRIMARY_CHAT_MODEL=gpt-4o-mini
OPENAI_FALLBACK_CHAT_MODEL=gpt-3.5-turbo
OPENAI_EMBEDDING_MODEL=text-embedding-3-small
OPENAI_HTTP_TIMEOUT_MS=90000
OPENAI_MAX_COMPLETION_TOKENS=8192

# Pinecone
PINECONE_API_KEY=pcsk_xxx
PINECONE_ENVIRONMENT=us-east-1
PINECONE_INDEX_NAME=gkchatty-sandbox

# Storage (choose one)
FILE_STORAGE_MODE=local|s3
# S3: AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, AWS_BUCKET_NAME, AWS_REGION
# Local: SYSTEM_KB_PATH, USER_DOCS_PATH

# Optional
REDIS_URL=redis://localhost:6379
ENCRYPTION_KEY=<64-char-hex>

```

Directory Structure

```

gkchatty-local/
├── backend/src/{config,controllers,middleware,models,routes,services,utils}/index.ts
├── frontend/src/{app,components,contexts,hooks,lib}/
└── ecosystem.config.js    # PM2
    └── netlify.toml        # Netlify
    └── .env                 # Config

```

PM2 Deployment

```

// ecosystem.config.js
module.exports = { apps: [
  { name: 'backend', script: './dist/index.js', cwd: './backend',
    instances: 4, exec_mode: 'cluster', max_memory_restart: '1G' },
  { name: 'frontend', script: 'npm', args: 'start', cwd: './frontend',
    instances: 2, exec_mode: 'cluster' }
]};

```

```

pm2 start ecosystem.config.js # Start
pm2 restart all             # Restart
pm2 logs                   # View logs
pm2 monit                  # Monitor

```

Build Commands

```

# Dev
cd backend && npm run dev
cd frontend && npm run dev

# Prod
cd backend && npm run build && node dist/index.js
cd frontend && npm run build && npm start

```

Database Models

Model	Collection	Key Fields
User	users	username, email, role, password
Chat	chats	userId, messages[], sources[]
UserDocument	userdocuments	userId, filename, s3Key
SystemKbDocument	systemkbdocuments	filename, textContent
Persona	personas	name, systemPrompt, isDefault
Folder	folders	name, userId, parentId
TenantKnowledgeBase	tenantkbs	name, users[]

Key API Endpoints

Endpoint	Method	Auth	Purpose
/health	GET	No	Health check
/api/auth/login	POST	No	Login
/api/chats	POST	User	Send message
/api/chats	GET	User	List chats
/api/documents/upload	POST	User	Upload doc
/api/admin/users	GET	Admin	List users
/api/admin/system-kb/upload	POST	Admin	Upload to System KB
/api/admin/reindex-system-kb	POST	Admin	Re-index vectors
/api/admin/server-info	GET	Admin	Server stats

Pinecone Namespaces

Namespace	Content
system-kb	System KB vectors
user-{userId}	Per-user vectors
tenant-{tenantId}	Tenant KB vectors

Rate Limits (Production)

Category	Limit
Standard	100/15min
Auth	20/min
AI/Chat	500/min
Upload	150/5min
Admin	200/15min

Security

- **Auth:** JWT in HTTP-only cookies, 24h expiry
- **Passwords:** bcrypt, 12 rounds
- **Headers:** Helmet.js (X-Frame-Options, XSS, nosniff)
- **CORS:** Dynamic origin from FRONTEND_URL
- **Rate Limiting:** Redis-backed (or in-memory)

Monitoring

```
# Health check
curl http://localhost:4001/health

# Logs
tail -f logs/app.log
pm2 logs --lines 100

# MongoDB
mongosh gkchatty --eval "db.stats()"
```

Common Issues

Issue	Check
Server won't start	lsof -i :4001, check MONGO_URI
MongoDB fail	mongosh --eval "db.adminCommand('ping')"
Pinecone errors	Verify API key, check index exists
OpenAI errors	Test key: curl api.openai.com/v1/models -H "Authorization: Bearer \$KEY"
Upload fails	Check disk space (df -h), file permissions

Maintenance

Task	Frequency	Command/Action
Check health	Daily	<code>curl /health</code>
Review logs	Daily	<code>pm2 logs --err</code>
Backup MongoDB	Weekly	<code>mongodump --uri=... --out=/backup/\$(date +%Y%m%d)</code>
Rotate API keys	Monthly	Update in Render/env
Update deps	Monthly	<code>npm audit fix</code>

Scaling (50+ users)

1. PM2: `instances: 'max'`
2. Enable Redis: `REDIS_URL=redis://host:6379`
3. MongoDB: `?maxPoolSize=200&minPoolSize=10`
4. Memory: `NODE_OPTIONS="--max-old-space-size=2048"`

Quick Commands

```
# Check ports
lsof -i :4001 && lsof -i :4003

# Restart services
pm2 restart all

# View env
printenv | grep -E "MONGO|OPENAI|PINECONE"

# MongoDB shell
mongosh "mongodb://localhost:27017/gkchatty"

# Test endpoints
curl -s localhost:4001/health | jq
```