

VIKING Technical Specification

VIKING Trading Analytics Dashboard - Technical Specification

1. EXECUTIVE SUMMARY
 2. ARCHITECTURE OVERVIEW
 3. TECHNOLOGY STACK
 - 3.1 Frontend
 - 3.2 Backend
 - 3.3 Database
 4. HOSTING OPTIONS
 - Option A: Single Server (Recommended for Simplicity)
 - Option B: Separate Services
 - Option C: Containerized (Docker)
 5. DNS CONFIGURATION (GoDaddy)
 - Required DNS Records
 6. ENVIRONMENT VARIABLES
 - 6.1 Frontend (.env)
 - 6.2 Backend (.env)
 7. API ENDPOINTS
 - Base URL: <https://api.getvng.com/api/viking>
 - Authentication
 8. DATABASE COLLECTIONS
 - Collections Used by VIKING
 - Database Access
 9. TRADING STRATEGIES
 - CORE Strategy
 - PRO Strategy
 10. FRONTEND COMPONENTS
 - Frontend Routes
 11. DEPLOYMENT CHECKLIST
 - Pre-Deployment
 - Deployment Steps
 - Post-Deployment Verification
 12. RECOMMENDED HOSTING PROVIDERS
 - Budget Option (~\$5-15/month)
 - Managed Option (~\$20-50/month)
 - Enterprise Option
 13. SECURITY CONSIDERATIONS
 - Immediate (Must Have)
 - Recommended
 - Future
 14. MONITORING & MAINTENANCE
 - Health Check Endpoint
 - Recommended Monitoring
 - Backup Strategy
 15. CONTACT & SUPPORT
- APPENDIX A: Quick Start Commands

VIKING Trading Analytics Dashboard - Technical Specification

Document Version: 1.0

Created: January 21, 2026

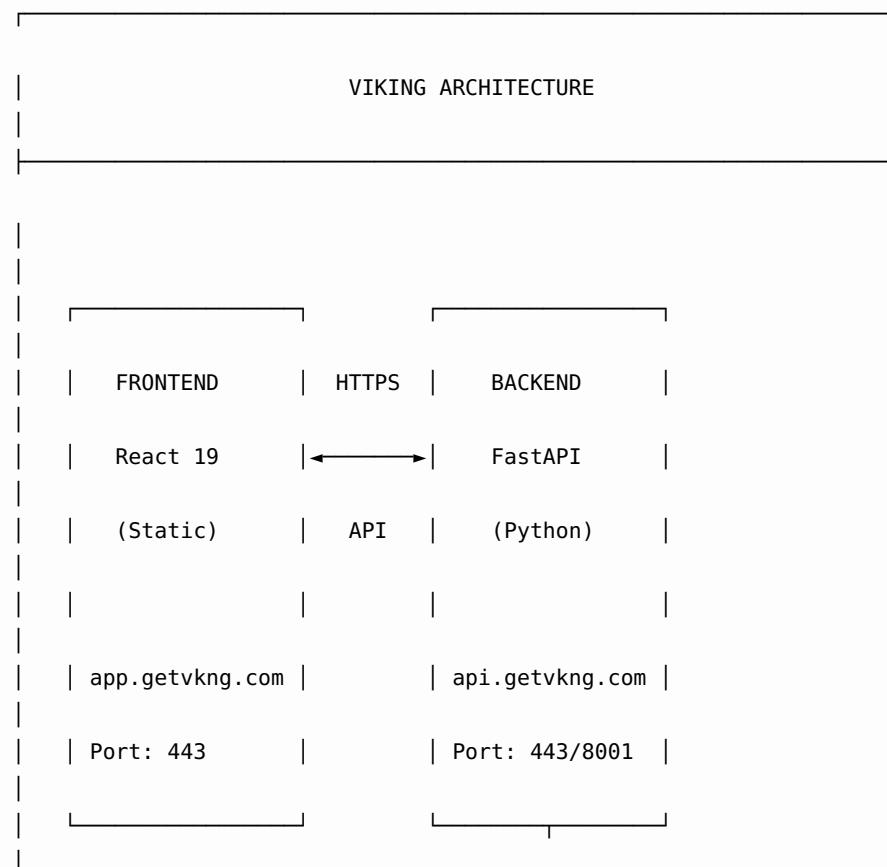
Purpose: Technical requirements for hosting VIKING on getvkng.com subdomain

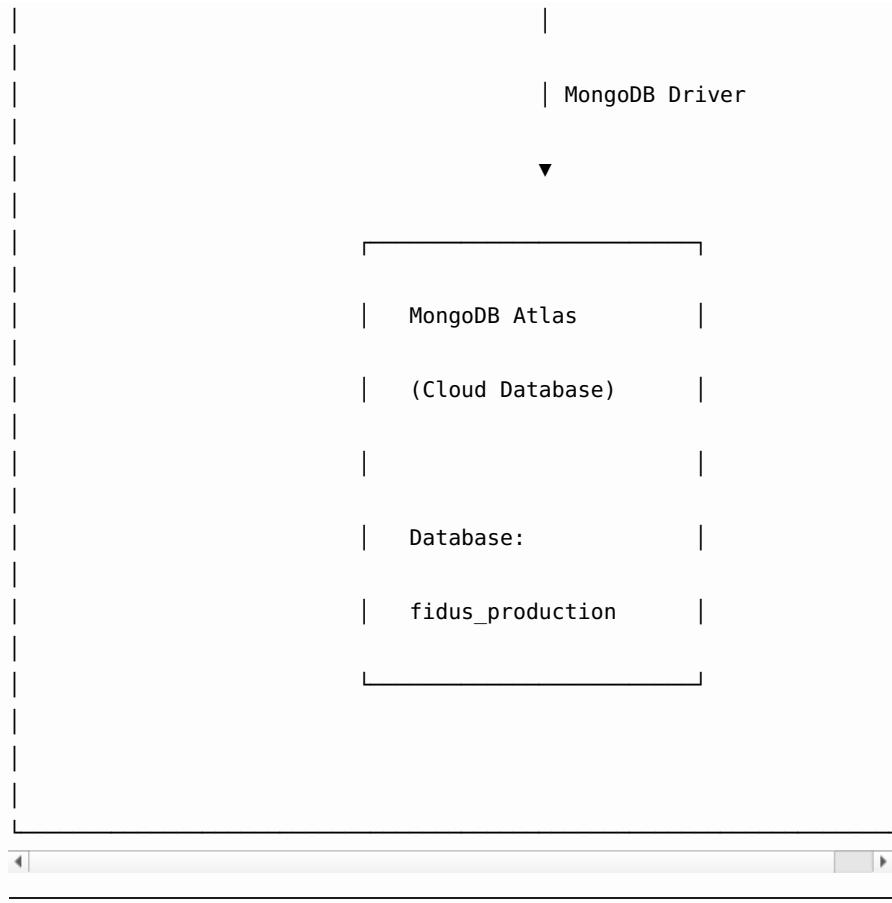
1. EXECUTIVE SUMMARY

VIKING is a standalone trading analytics dashboard that displays performance metrics for two trading strategies (CORE and PRO). It is currently part of the FIDUS platform but is designed to be deployed as a separate application.

Recommended Subdomain: app.getvkng.com or dashboard.getvkng.com

2. ARCHITECTURE OVERVIEW





3. TECHNOLOGY STACK

3.1 Frontend

Component	Version	Notes
React	19.0.0	Main framework
Tailwind CSS	3.x	Styling
Recharts	Latest	Charts/graphs
Lucide React	Latest	Icons
Axios	Latest	HTTP client

Build Output: Static files (HTML, JS, CSS)

Build Command: `npm run build` or `yarn build`

Output Directory: `/build`

3.2 Backend

Component	Version	Notes
Python	3.11+	Runtime
FastAPI	0.110.1	Web framework
Motor	Latest	Async MongoDB driver
Uvicorn	Latest	ASGI server
PyJWT	Latest	JWT authentication

Start Command: uvicorn server:app --host 0.0.0.0 --port 8001

3.3 Database

Component	Details
Provider	MongoDB Atlas
Database Name	fidus_production
Connection	MongoDB SRV (TLS encrypted)

4. HOSTING OPTIONS

Option A: Single Server (Recommended for Simplicity)

```
app.getvkng.com
├── Frontend (Nginx serving static files)
├── Backend (Uvicorn/Gunicorn on port 8001)
└── Nginx reverse proxy for /api/* → localhost:8001
```

Option B: Separate Services

```
app.getvkng.com      → Frontend (Static hosting: Netlify, Vercel,
Cloudflare Pages)
api.getvkng.com     → Backend (VPS, Render, Railway, DigitalOcean
App Platform)
```

Option C: Containerized (Docker)

```
docker-compose.yml with:
├── frontend (nginx:alpine)
├── backend (python:3.11-slim)
└── nginx (reverse proxy)
```

5. DNS CONFIGURATION (GoDaddy)

Required DNS Records

Type	Name	Value	TTL
A	app	[Server IP]	600
A	api	[Server IP]	600
CNAME	www.app	app.getvkng.com	600

If using cloud hosting (Render, Vercel, etc.): | Type | Name | Value | TTL | |---|---|---|---| | CNAME | app | [provider-domain].com | 600 |

6. ENVIRONMENT VARIABLES

6.1 Frontend (.env)

```
# Required
REACT_APP_BACKEND_URL=https://api.getvkng.com

# Optional
REACT_APP_APP_NAME=VIKING
```

6.2 Backend (.env)

```
# Required - Database
MONGO_URL=mongodb+srv://[username]:[password]@[cluster].mongodb.net/fidus_production?
    retryWrites=true&w=majority

DB_NAME=fidus_production

# Required - Security
JWT_SECRET_KEY=[generate-secure-random-string]

# Required - CORS
FRONTEND_URL=https://app.getvkng.com
BACKEND_URL=https://api.getvkng.com

# Optional - MT5 Bridge (for live data sync)
MT5_BRIDGE_URL=http://[vps-ip]:8000
MT5_BRIDGE_API_KEY=[bridge-api-key]
MT5_BRIDGE_ACTIVE=false
```

7. API ENDPOINTS

Base URL: <https://api.getvkng.com/api/viking>

Method	Endpoint	Description
GET	/accounts	Get all VIKING accounts (CORE & PRO)
GET	/accounts/{account_number}	Get specific account
GET	/deals/{strategy}	Get trade history (CORE or PRO)
GET	/analytics/{strategy}	Get performance analytics
GET	/summary	Get combined summary
GET	/symbols/{strategy}	Get symbol distribution
GET	/balance-history/{strategy}	Get balance chart data
GET	/monthly-returns/{strategy}	Get monthly returns
GET	/risk/{strategy}	Get risk analysis

POST	/calculate-analytics/{strategy}	Recalculate analytics
------	---------------------------------	-----------------------

Authentication

Currently uses simple username/password authentication: -

Username: admin - **Password:** Password123

Recommendation: Implement proper JWT authentication for production.

8. DATABASE COLLECTIONS

Collections Used by VIKING

Collection	Purpose	Records
mt5_accounts	Master account data (shared with FIDUS)	~15
mt5_deals	MT5 trade history	~3,000+
viking_accounts	VIKING-specific account metadata	~3
viking_deals_history	MT4 historical trades	~4,700
viking_analytics	Calculated performance metrics	~4
viking_balance_history	Balance snapshots for charts	~1,600

Database Access

VIKING shares the fidus_production database with FIDUS. For a completely isolated deployment, you can: 1. Export VIKING collections to a new database 2. Update DB_NAME environment variable 3. Run initial data seeding

9. TRADING STRATEGIES

CORE Strategy

Field	Value
Current Account	885822 (MT5)
Historical Account	33627673 (MT4)
Platform	MT5
Broker	MEXAtlantic
Total Trades	~4,143

PRO Strategy

Field	Value
Account	1309411
Platform	MT4
Broker	Traders Trust
Total Trades	~1,152

10. FRONTEND COMPONENTS

File	Lines	Purpose
VikingApp.js	214	Main app wrapper, routing
VikingLogin.js	293	Authentication page
VikingDashboard.js	1,609	Main dashboard (Overview, Orders, Analytics)

Frontend Routes

Path	Component	Description
/viking	VikingApp	Entry point
/viking/login	VikingLogin	Login page
/viking/dashboard	VikingDashboard	Main dashboard

11. DEPLOYMENT CHECKLIST

Pre-Deployment

- MongoDB Atlas: Whitelist server IP or allow all (0.0.0.0/0)
- Generate secure JWT_SECRET_KEY (min 32 characters)
- SSL certificate for domain (Let's Encrypt recommended)
- Configure CORS origins in backend

Deployment Steps

1. DNS Setup

- Add A/CNAME records in GoDaddy
- Wait for propagation (5-30 minutes)

2. Backend Deployment

```
# Clone repository
git clone [repo-url]
cd backend

# Install dependencies
```

```

pip install -r requirements.txt

# Set environment variables
cp .env.example .env
# Edit .env with production values

# Start server
uvicorn server:app --host 0.0.0.0 --port 8001

```

3. Frontend Deployment

```

cd frontend

# Set environment variables
echo "REACT_APP_BACKEND_URL=https://api.getvkng.com" > .env

# Build
npm install
npm run build

# Deploy /build folder to web server

```

4. Nginx Configuration (if using single server)

```

server {
    listen 443 ssl;
    server_name app.getvkng.com;

    ssl_certificate /path/to/cert.pem;
    ssl_certificate_key /path/to/key.pem;

    # Frontend
    location / {
        root /var/www/viking/build;
        try_files $uri $uri/ /index.html;
    }

    # Backend API proxy
    location /api/ {
        proxy_pass http://localhost:8001/api/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

Post-Deployment Verification

- Frontend loads at https://app.getvkng.com/viking
 - API responds at https://api.getvkng.com/api/viking/accounts
 - Login works with admin/Password123
 - Both CORE and PRO strategies display data
 - Charts render correctly
 - No console errors in browser
-

12. RECOMMENDED HOSTING PROVIDERS

Budget Option (~\$5-15/month)

- **DigitalOcean Droplet** (\$6/mo) + Nginx
- **Hetzner Cloud** (\$4/mo) + Nginx
- **Vultr** (\$5/mo) + Nginx

Managed Option (~\$20-50/month)

- **Render.com** - Backend as Web Service, Frontend as Static Site
- **Railway.app** - Full stack deployment
- **DigitalOcean App Platform** - Containerized deployment

Enterprise Option

- **AWS** - EC2 + CloudFront + RDS (if migrating from MongoDB)
 - **Google Cloud** - Cloud Run + Cloud CDN
 - **Azure** - App Service + CDN
-

13. SECURITY CONSIDERATIONS

Immediate (Must Have)

- HTTPS only (redirect HTTP to HTTPS)
- Strong JWT secret key
- MongoDB connection via SRV with TLS
- Rate limiting on API endpoints

Recommended

- Replace hardcoded credentials with proper user management
- Implement refresh tokens
- Add API key authentication for sensitive endpoints
- Enable MongoDB audit logging

Future

- Two-factor authentication
 - IP whitelisting for admin access
 - Web Application Firewall (WAF)
-

14. MONITORING & MAINTENANCE

Health Check Endpoint

```
GET /api/health
```

Recommended Monitoring

- **Uptime:** UptimeRobot, Pingdom (free tier available)
- **Error Tracking:** Sentry (free tier available)
- **Logs:** Logtail, Papertrail, or server logs

Backup Strategy

- MongoDB Atlas provides automatic daily backups
- Export critical data weekly to separate storage

15. CONTACT & SUPPORT

For technical questions about VIKING implementation:
- Review this document first
- Check MongoDB Atlas connection and IP whitelist
- Verify environment variables are set correctly

APPENDIX A: Quick Start Commands

```
# Backend
cd /app/backend
pip install -r requirements.txt
export MONGO_URL="mongodb+srv://..."
export DB_NAME="fidus_production"
export JWT_SECRET_KEY="your-secret-key"
export FRONTEND_URL="https://app.getvkng.com"
uvicorn server:app --host 0.0.0.0 --port 8001

# Frontend
cd /app/frontend
npm install
REACT_APP_BACKEND_URL=https://api.getvkng.com npm run build
# Serve /build folder with any static file server
```

APPENDIX B: Docker Compose (Optional)

```
version: '3.8'
services:
  frontend:
    build: ./frontend
    ports:
      - "3000:80"
    environment:
      - REACT_APP_BACKEND_URL=http://backend:8001
    depends_on:
      - backend
```

```
backend:  
  build: ./backend  
  ports:  
    - "8001:8001"  
environment:  
  - MONGO_URL=${MONGO_URL}  
  - DB_NAME=fidus_production  
  - JWT_SECRET_KEY=${JWT_SECRET_KEY}  
  - FRONTEND_URL=https://app.getvkng.com  
  
nginx:  
  image: nginx:alpine  
  ports:  
    - "80:80"  
    - "443:443"  
  volumes:  
    - ./nginx.conf:/etc/nginx/nginx.conf  
    - ./certs:/etc/nginx/certs  
  depends_on:  
    - frontend  
    - backend
```

Document End