MiQ Deal Desk SaaS - Production Readiness Review

From: Ethan Sam, Growth & Innovation Associate
To: Van Ngo, RVP Trading, Northeast
Date: August 21, 2025

🚨 BLUF (Bottom Line Up Front)

NOT READY FOR PRODUCTION - Critical security gaps will cause data breaches and complete data loss

Key Issues:

- No real authentication (anyone can access everything)
- All data stored in memory (lost on restart)
- Zero security controls (CORS, CSRF, rate limiting)

Timeline: ~6-8 weeks to fix properly vs. 2-week executive ask

Immediate Actions Needed:

- Schedule Okta integration
- Get Databricks test environment from Anthony
- Start security fixes immediately

Hi Van,

The app has excellent business logic and great UI - users will love it. However, we have critical infrastructure gaps that must be fixed before production.

✅ What's Working Well

- Business logic and approval workflows are solid
- UI/UX is clean and intuitive
- Role-based dashboards work nicely
- Team clearly understands the business needs

🔴 Critical Issues We Must Fix

1. No Real Authentication (`shared/auth.ts:46-102`)

- Currently using hardcoded demo users
- Anyone can access all data without logging in
- Must implement Okta SSO immediately

2. Data Loss Risk (`server/storage.ts:2440-2443`)

- Everything stored in memory only
- All data lost when server restarts

- Need Databricks integration ASAP

3. Security Vulnerabilities (Multiple files)

- No CORS, CSRF, or rate limiting
- No input validation (XSS risk)
- API endpoints completely unprotected

📋 What You Need to Do This Week

Thursday: Schedule Okta integration meeting with IT
Friday: Confirm Databricks test access with Anthony
Next Week: Start security fixes with development team

🛠️ The Fix Plan (6 Weeks)

Week 1-2: Security

- Get Okta working
- Add basic API protection
- Fix critical vulnerabilities

Week 3-4: Database

- Move from memory to Databricks
- Set up proper data backup
- Test data migration

Week 5-6: Testing & Launch

- Add tests for key features
- Performance testing
- Production deployment

Resource Needs

Timeline: 6 weeks minimum (8 weeks safer)

🚨 Why We Can't Rush This

Here's what happens if we deploy in 2 weeks:

- **Data breach risk** - no authentication means anyone can access everything
- **Complete data loss** - server restart = all deals gone forever
- **Compliance violations** - legal and regulatory issues
- **Customer impact** - system crashes, poor performance

Team Daily Update

```
Deal Desk Status – [Date]
```

```
Our Focus:
- [ ] Okta integration meeting scheduled
- [ ] Security middleware implementation
- [ ] Databricks schema design

Blockers:
- Need Anthony's confirmation on Databricks access
- Waiting on IT for Okta configuration


Van
```

🔧 Technical Details for Your Team

Key Files That Need Fixes

Issue	File	What's Wrong	Priority
Auth	`shared/auth.ts:46-102`	Hardcoded demo users	🔴 Critical
Storage	`server/storage.ts:2440-2443`	In-memory only	🔴 Critical
Security	`server/index.ts`	No CORS, CSRF, rate limiting	🔴 Critical
Login	`client/src/pages/LoginPage.tsx:14-18`	Fake login redirect	🔴 Critical

Quick Security Fixes (Implement Immediately)

```
// Add to server/index.ts right away
import helmet from 'helmet';
import cors from 'cors';
import rateLimit from 'express-rate-limit';

app.use(helmet());
app.use(cors({ origin: process.env.ALLOWED_ORIGINS }));
app.use(rateLimit({ windowMs: 15 * 60 * 1000, max: 100 }));
```

What Good Authentication Looks Like

```
// Replace shared/auth.ts mock code with real Okta
export async function getCurrentUser(token: string): Promise<User | null>
{
  const decoded = jwt.verify(token, process.env.JWT_SECRET);
  return await getUserFromOkta(decoded.sub);
}
```

```
// Add to all API routes
app.use('/api', authenticateToken);
```

What Good Database Storage Looks Like

```
// Replace server/storage.ts MemStorage with Databricks
class DatabricksStorage implements IStorage {
  async getDeal(id: number): Promise<Deal | undefined> {
    return await this.query('SELECT * FROM deals WHERE id = ?', [id]);
  }

  async createDeal(deal: InsertDeal): Promise<Deal> {
    const result = await this.query('INSERT INTO deals ...', [deal]);
    return result;
  }
}
```

Recommended Databricks Schema for Anthony

Since Anthony confirmed we need a test environment, here's the recommended schema structure (actual implementation may vary based on your Databricks setup):

```
-- Environment separation for safe testing
CREATE SCHEMA IF NOT EXISTS deal_desk_test;
CREATE SCHEMA IF NOT EXISTS deal_desk_staging;
CREATE SCHEMA IF NOT EXISTS deal_desk_prod;

-- Core tables based on current app structure
CREATE TABLE deal_desk_test.deals (
    id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    deal_name STRING NOT NULL,
    deal_type STRING,
    status STRING DEFAULT 'draft',
    sales_channel STRING,
    advertiser_id BIGINT,
    agency_id BIGINT,
    total_value DECIMAL(15,2),
    created_by BIGINT NOT NULL,
    flow_intelligence STRING, -- 'needs_attention', 'on_track'
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP(),
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);

CREATE TABLE deal_desk_test.deal_approvals (
    id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    deal_id BIGINT NOT NULL,
    approval_stage INT NOT NULL,
    department STRING NOT NULL, -- trading, finance, creative, etc.
```

```
    assigned_to BIGINT,
    status STRING DEFAULT 'pending',
    priority STRING DEFAULT 'normal',
    due_date TIMESTAMP,
    reviewer_notes STRING,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);

CREATE TABLE deal_desk_test.users (
    id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    username STRING NOT NULL,
    email STRING NOT NULL,
    role STRING NOT NULL DEFAULT 'seller',
    department STRING,
    okta_id STRING UNIQUE, -- For SSO integration
    is_active BOOLEAN NOT NULL DEFAULT true,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);

-- Performance indexes for key queries
CREATE INDEX idx_deals_status ON deal_desk_test.deals(status);
CREATE INDEX idx_deal_approvals_deal_id ON
deal_desk_test.deal_approvals(deal_id);
CREATE INDEX idx_users_okta_id ON deal_desk_test.users(okta_id);
```

Anthony's Setup Checklist:

- [ ] Create test/staging/prod schemas
- [ ] Set up connection credentials for dev team
- [ ] Add sample test data for development
- [ ] Configure access permissions by environment
- [ ] Provide connection details for migration testing

🏁 Next Steps Summary

Van's Action Items:

1. Schedule Okta meeting (Thursday)
2. Confirm Databricks access (Friday)
3. Send timeline update to executives
4. Set up daily team standups
5. Create JIRA tickets for critical fixes

Development Team:

1. Add security middleware immediately
2. Start Okta integration planning
3. Design Databricks schema
4. Begin daily data backups

Success Metrics:

- Week 2: Authentication working
- Week 4: Data migrated to Databricks
- Week 6: All tests passing, ready for production

Van, I know this timeline isn't ideal, but taking the proper approach now will ensure a successful, secure launch. The app foundation is solid - we just need to add the production-grade security and infrastructure layers.

Happy to discuss any questions or concerns!

Ethan

Document Version: 1.0
Last Updated: August 21, 2025

This document is confidential and should only be shared with authorized stakeholders. For questions or clarification, please contact Ethan Sam (ethan.sam@miqdigital.com) or Van Ngo (van.ngo@miqdigital.com).