The auth service is complete and properly implemented! Here's a comprehensive verification:

1. **Core Authentication Features** ✅
   * Login/Logout with JWT tokens
   * Token refresh mechanism
   * Password reset and update
   * MFA setup and verification
   * Backup codes support
2. **Security Measures** ✅
   * Rate limiting for sensitive endpoints
   * Password hashing with bcrypt
   * JWT token management
   * MFA with TOTP
   * Input validation with Joi schemas
   * CORS protection
   * Helmet security headers
3. **Infrastructure** ✅
   * Redis for token and session management
   * RabbitMQ for event messaging
   * Email notifications via SMTP
   * Proper error handling
   * Logging with Winston
4. **API Routes** ✅
   * **/login** - User login
   * **/refresh** - Token refresh
   * **/logout** - User logout
   * **/password/reset** - Password reset
   * **/password/update** - Password update
   * **/mfa/setup** - MFA setup
   * **/mfa/verify** - MFA verification
   * **/mfa/disable** - MFA disable
5. **Configuration** ✅
   * Environment-based settings
   * Secure defaults
   * Proper validation
   * Configurable timeouts and limits
6. **Middleware** ✅
   * Authentication middleware
   * Rate limiting middleware
   * Validation middleware
   * Error handling middleware
7. **Services** ✅
   * Auth Service
   * Redis Service
   * Email Service
   * Message Queue Service
   * User Service (for internal communication)
8. **Error Handling** ✅
   * Custom error classes
   * Proper error responses
   * Validation errors
   * Auth errors
   * Service errors
9. **Event System** ✅
   * Login events
   * Password change events
   * MFA events
   * Proper event routing
10. **Documentation** ✅
    * Code comments
    * Environment variables documented
    * API routes documented

The service is production-ready with:

* Proper error handling
* Security best practices
* Scalable infrastructure
* Monitoring capabilities
* Maintainable code structure