 User Service is now complete with all necessary features implemented. Here's a comprehensive overview:

1. Core User Management: ✅ CRUD operations ✅ Profile management ✅ Role management ✅ Status management ✅ Bulk operations
2. Security Features: ✅ Password hashing with bcrypt ✅ MFA setup and verification ✅ Backup codes generation ✅ Email verification ✅ Account locking ✅ Password history ✅ Session management
3. Database Models: ✅ User model with all fields ✅ Profile model ✅ Session model ✅ Role model ✅ Proper indexes and relationships
4. API Routes: ✅ User management endpoints ✅ Profile endpoints ✅ Security endpoints (MFA, backup codes) ✅ Session management endpoints ✅ Account management endpoints
5. Validation & Error Handling: ✅ Input validation with Joi ✅ Error handling middleware ✅ Custom error types ✅ Request validation
6. Caching & Performance: ✅ Redis caching ✅ Cache invalidation ✅ Efficient queries ✅ Proper indexing
7. Event System: ✅ RabbitMQ integration ✅ Event publishing ✅ Async operations
8. Security Middleware: ✅ Authentication ✅ Permission checking ✅ Rate limiting ✅ Input sanitization

The service is production-ready with:

* Proper error handling
* Input validation
* Security features
* Performance optimizations
* Scalability considerations
* Audit logging
* Event-driven architecture

To start using it, you need to:

1. Run the database migrations
2. Configure environment variables
3. Set up Redis and RabbitMQ connections
4. Define roles and permissions
5. Set up email service for verification