

Database Migration Complete - Authentication System

Overview

Successfully migrated the Stock Picking Agent to a full authentication system with user-specific portfolios.

What Was Implemented

1. Authentication System

- **NextAuth.js v4** with Prisma adapter
- Credential-based login (email + password)
- Password hashing with bcrypt
- JWT session management
- User registration with automatic portfolio creation

2. Database Schema Updates

New Models Added:

- **User:** Email, password, name
- **Account:** OAuth account linking (future use)
- **Session:** Session tracking
- **VerificationToken:** Email verification (future use)

Portfolio Model Enhanced:

- Added `userId` field for user ownership
- Added `isLocked` field for modification control
- Added unique constraint on `[userId, name]`

Stock Model Enhanced:

- Added `rating` field for user ratings
- Added `ratingUpdatedAt` timestamp
- Linked to portfolios via `portfolioId`

3. API Routes Updated

All portfolio and stock API routes now:

- Verify user authentication
- Check portfolio ownership
- Filter data by user
- Return appropriate error codes

4. UI Components

New Pages:

- `/login` - User login page
- `/signup` - User registration page

Updated Components:

- `SidebarNavigation` - Shows user profile and sign-out button
- `LayoutContent` - Conditionally renders navigation based on auth state
- `DashboardClient` - Displays user-specific data

5. Middleware

Protects all routes except:

- `/login`
- `/signup`
- `/api/auth/*`
- `/api/signup`

Test Users Created

Demo User

- **Email:** `demo@portfolio-intelligence.co.uk`
- **Password:** `demo123456`
- **Portfolios:** Default portfolios created automatically

Test User

- **Email:** `john@doe.com`
- **Password:** `johndoe123`
- **Portfolios:** Default portfolios created automatically

Build Issues Resolved

Issue 1: Database Connection During Build

Problem: Build command tried to run `prisma db push` during Vercel build


Solution: Removed database operations from build command

Status:  Fixed

Issue 2: Yarn.lock Symlink

Problem: `yarn.lock` was a symlink that Vercel couldn't resolve

Solution: Replaced with actual 515KB file

Status:  Fixed (recurring issue documented in `YARN_LOCK_ISSUE.md`)

Issue 3: TypeScript `userId` Errors

Problem: Prisma queries didn't explicitly select `userId` field

Solution: Added explicit select statements for `userId` in all portfolio queries

Status:  Fixed

Issue 4: Prisma Client Not Regenerated

Problem: TypeScript couldn't find `userId` in `Portfolio` type

Solution: Ran `yarn prisma generate` to regenerate Prisma client

Status:  Fixed

Current Build Status

- ✓ **TypeScript Compilation:** PASSING
- ✓ **Production Build:** SUCCESSFUL
- ✓ **All API Routes:** FUNCTIONAL
- ✓ **Authentication:** WORKING
- ✓ **User-Specific Data:** WORKING

Database Commands

Generate Prisma Client

```
cd nextjs_space  
yarn prisma generate
```

Push Schema Changes

```
cd nextjs_space  
yarn prisma db push
```

Create Migration

```
cd nextjs_space  
yarn prisma migrate dev --name description_of_changes
```

Seed Database

```
cd nextjs_space  
yarn prisma db seed
```

Environment Variables Required

Development (.env)

```
DATABASE_URL="postgresql://..."  
NEXTAUTH_SECRET="your-secret-key"  
NEXTAUTH_URL="http://localhost:3000"
```

Production (Vercel)

```
DATABASE_URL="postgresql://..."  
NEXTAUTH_SECRET="your-secret-key"  
NEXTAUTH_URL="https://portfolio-intelligence.co.uk"
```

Vercel Deployment Checklist

- ✓ Environment variables configured
- ✓ yarn.lock is a regular file (not symlink)

- ☒ Build command doesn't include database operations
- ☒ Prisma client regenerates via postinstall script
- ☒ Database schema is up to date
- ☒ All TypeScript errors resolved

Key Files Modified

Database

- `prisma/schema.prisma` - Added User, Portfolio, Stock fields
- `scripts/seed.ts` - Creates test users and portfolios

Authentication

- `lib/auth.ts` - NextAuth configuration
- `app/api/auth/[...nextauth]/route.ts` - Auth endpoints
- `app/api/signup/route.ts` - User registration
- `app/login/page.tsx` - Login page
- `app/signup/page.tsx` - Signup page
- `middleware.ts` - Route protection

API Routes (User-Aware)

- `app/api/portfolios/route.ts` - List/create portfolios
- `app/api/portfolios/[id]/route.ts` - Update/delete portfolios
- `app/api/portfolios/[id]/stocks/[stockId]/route.ts` - Stock operations
- `app/api/stock/move/route.ts` - Move stocks between portfolios
- `app/api/stock/update-rating/route.ts` - Update stock ratings

UI Components

- `components/layout-content.tsx` - Conditional navigation
- `components/sidebar-navigation.tsx` - User profile display
- `app/providers.tsx` - SessionProvider wrapper
- `app/layout.tsx` - Root layout with providers

Testing the System

1. Test Signup

```
# Visit http://localhost:3000/signup
# Create new account
# Verify redirect to dashboard
# Check portfolios are created
```

2. Test Login

```
# Visit http://localhost:3000/login
# Use demo@portfolio-intelligence.co.uk / demo123456
# Verify redirect to dashboard
# Check user-specific data loads
```

3. Test Authorization

```
# Try accessing /dashboard without login
# Should redirect to /login
# Login and verify access granted
```

4. Test Portfolio Operations

```
# Create new portfolio
# Add stocks to portfolio
# Switch between portfolios
# Verify data isolation between users
```

Known Issues

Yarn.lock Symlink Reversion

Issue: yarn.lock reverts to symlink after checkpoint operations

Impact: Breaks Vercel builds

Workaround: Replace with actual file before each push

Documentation: See YARN_LOCK_ISSUE.md

Session Provider on Auth Pages

Issue: SessionProvider caused hydration errors on login/signup pages

Solution: Conditional rendering in LayoutContent component

Status:  Resolved

Next Steps

Optional Enhancements

1. **Email Verification:** Implement email verification for new users
2. **Password Reset:** Add forgot password functionality
3. **OAuth Providers:** Add Google/GitHub login options
4. **User Profile:** Add profile editing page
5. **Portfolio Sharing:** Allow users to share portfolios

Performance Optimizations


1. **Caching:** Implement Redis for session caching
2. **Pagination:** Add pagination for large portfolios
3. **Lazy Loading:** Implement lazy loading for stock data
4. **Database Indexes:** Optimize database queries with indexes

Support

For issues or questions:

1. Check VERCEL_BUILD_FIX.md for deployment issues
2. Check YARN_LOCK_ISSUE.md for symlink problems
3. Check DATABASE_MIGRATION_COMPLETE.md (this file) for migration details

Migration Completed: December 10, 2024

Status:  Fully Functional

Next Action: Monitor Vercel deployment