

MongoDB Integration Complete! 🎉

☑ What I've Built

Database Models (MongoDB + Mongoose)

- **User**: Authentication with bcrypt password hashing
- **MoodEntry**: User-specific mood tracking with date/score/notes
- **PersonalityAssessment**: Big Five results with full history
- **ChatHistory**: Conversation persistence per user/session

Authentication System

- JWT-based auth with 7-day token expiration
- Secure registration (`POST /api/auth/register`)
- Login with credential validation (`POST /api/auth/login`)
- Protected routes middleware for all user data endpoints

Updated Services

- `moodService`: Now saves to MongoDB with `userId`
- `assessmentService`: Stores personality results per user
- All endpoints require authentication except public resources

New Auth-Enabled Frontend

- Login/Register modal with form validation
- JWT token storage in `localStorage`
- Automatic auth header injection for all API calls
- User display with logout functionality
- Content hidden until authenticated

Files Created/Modified

Models:

- `models/User.js`
- `models/MoodEntry.js`
- `models/PersonalityAssessment.js`
- `models/ChatHistory.js`

Config:

- `config/database.js` (MongoDB connection)
- `middleware/auth.js` (JWT middleware)

Updated:

- `server.js` (auth routes, protected endpoints)
- `services/moodService.js` (MongoDB integration)
- `services/assessmentService.js` (MongoDB integration)
- `public/index.html` (auth UI)
- `public/app.js` (auth logic)
- `public/styles.css` (modal styles)

Documentation:

- `README-NEW.md` (comprehensive MongoDB setup guide)
- `.env.example` (MongoDB URI, JWT secret)

Next Steps to Run

1. Install MongoDB (if not already installed)

Windows - MongoDB Community Server:

```
# Download from: https://www.mongodb.com/try/download/community
# Run installer, use defaults
# MongoDB starts as Windows service automatically
```

Verify MongoDB is running:

```
Get-Service MongoDB
# Should show "Running" status
```

2. Create .env File

```
Copy-Item .env.example .env
```

Edit `.env` and set at minimum:

```
MONGODB_URI=mongodb://localhost:27017/mindbot
JWT_SECRET=change-this-to-something-secure-and-random
```

3. Start Server

```
# Stop any running node processes
Get-Process node -ErrorAction SilentlyContinue | Stop-Process -Force
```

```
# Start server
npm run dev
```

4. Open Browser

Navigate to: **http://localhost:3000**

5. Test Authentication Flow

1. Click **Register** button
2. Enter:
 - Username: **testuser**
 - Email: **test@example.com**
 - Password: **password123**
3. Click Register
4. You should be logged in automatically
5. Try logging mood, taking personality test
6. Check MongoDB to see data:

```
mongosh
use mindbot
db.users.find()
db.moodentries.find()
```

Key Features

- ✓ **User-specific data:** Each user's mood/assessments isolated by userId
- ✓ **Secure auth:** Passwords hashed with bcrypt, JWT tokens expire after 7 days
- ✓ **Protected endpoints:** All user data requires authentication
- ✓ **Persistent storage:** No more JSON files, all data in MongoDB
- ✓ **History tracking:** View past personality assessments and mood trends
- ✓ **Clean UI:** Login/register modal, username display, logout button

Database Collections

After registration and some usage:

- **users** - User accounts
- **moodentries** - Daily mood logs per user
- **personalityassessments** - Big Five results with timestamps
- **chathistories** - AI conversation sessions (ready for future use)

Production Considerations

Before going live:

1. Change `JWT_SECRET` to strong random value
2. Use MongoDB Atlas or managed service
3. Add HTTPS/TLS
4. Implement rate limiting
5. Add email verification
6. Enable MongoDB encryption

Quick Troubleshooting

"MongoDB connection error"

- Check if MongoDB service is running
- Verify `MONGODB_URI` in `.env`

"Authentication required" errors

- Clear localStorage: F12 → Console → `localStorage.clear()`
- Re-register/login

Frontend not showing auth buttons

- Hard refresh: Ctrl+Shift+R
- Check console for JS errors

Ready to test! Open <http://localhost:3000> and register your first user! 🚀