Database Setup Summary

Date: October 23, 2025

Issue Resolved

✅ **Fixed: “The table `public.User` does not exist in the current database”**

What Was Done

1. Schema Verification

- Reviewed `prisma/schema.prisma` - 20+ tables defined
- Confirmed database schema structure
- Verified binary targets for Vercel compatibility

2. Migration Analysis

- Found 2 existing migrations:
- `20251020102014_init` - Initial schema
- `20251020102138_update_mode_schema` - Mode updates
- Confirmed PostgreSQL as the database provider

3. Prisma Client Generation

```
npx prisma generate
```

✅ Successfully generated Prisma Client v6.7.0

4. Database Baseline

The database already had tables but lacked migration tracking. Used baseline approach:

```
npx prisma migrate resolve --applied "20251020102014_init"
npx prisma migrate resolve --applied "20251020102138_update_mode_schema"
```

This created the `_prisma_migrations` table and marked existing migrations as applied.

5. Migration Deployment

```
npx prisma migrate deploy
```

✅ Result: No pending migrations to apply

6. Configuration Updates

Updated `package.json` with new scripts:

```
{
  "scripts": {
    "migrate:deploy": "prisma migrate deploy",
    "db:push": "prisma db push",
    "db:studio": "prisma studio"
  }
}
```

7. Database Verification

Created and ran verification script:

```
✓ Database connection successful
✓ Found 21 tables (including _prisma_migrations)
✓ Found 2 applied migrations
✓ User table accessible (2 users)
✓ ClientProfile table accessible (2 profiles)
✓ Engagement table accessible (2 engagements)
✓ AssessmentImport table accessible (0 imports)
```

Database Tables

All 21 tables are now properly tracked and verified:

1. Account
2. AssessmentImport
3. AuditEvent
4. ClientProfile
5. CoachingNote
6. CoachingSession
7. ComputedResult
8. Engagement
9. MappingVersion
10. Mode
11. Organization
12. PasswordResetToken
13. Plan
14. PlanItem
15. SchemaPackVersion
16. SchemaResolution
17. Session
18. SessionNote
19. User
20. VerificationToken
21. _prisma_migrations (system table)

Documentation Created

1. **DATABASE_SETUP.md** - Comprehensive database setup guide
   - Initial setup instructions
   - Migration management
   - Troubleshooting
   - Best practices

2. **DEPLOYMENT_CHECKLIST.md** - Step-by-step deployment guide
   - Pre-deployment steps
   - Deployment process
   - Verification steps
   - Rollback plan

3. **DATABASE_SETUP_SUMMARY.md** - This document

Database Connection

Current Status: ✅ Connected and Verified

```
Host: db-1035c2da98.db002.hosteddb.reai.io
Port: 5432
Database: 1035c2da98
Schema: public
```

Next Steps

1. **Commit changes** ✅ (Next)
2. **Push to GitHub** - deployment-fix-verified branch
3. **Verify in production** - Test the deployed application
4. **Monitor** - Check logs after deployment

Files Modified

- `package.json` - Added database scripts
- `README_DEPLOYMENT.md` - Updated documentation index

Files Created

- `docs/DATABASE_SETUP.md`
- `docs/DEPLOYMENT_CHECKLIST.md`
- `docs/DATABASE_SETUP_SUMMARY.md`
- `verify_database.js` (temporary, for testing)

Verification Scripts

Created helper scripts for database verification:
- `check_db.js` - Check existing tables

- `check_migrations.js` - Check migration status
- `verify_database.js` - Comprehensive verification

Key Learnings

1. **Baseline Existing Databases**: When tables exist without migration history, use `prisma migrate resolve --applied` to baseline
2. **Vercel Binary Targets**: Ensure `binaryTargets` includes `"linux-musl-openssl-3.0.x"` for Vercel
3. **Postinstall Script**: The existing `postinstall: prisma generate` script ensures Prisma Client is generated on Vercel builds
4. **Migration Order**: Always deploy migrations before deploying code changes

Production Readiness

✅ **Database**: Fully configured and verified
✅ **Migrations**: All applied and tracked
✅ **Prisma Client**: Generated and ready
✅ **Documentation**: Comprehensive guides created
✅ **Scripts**: Helper commands added to package.json

Status: READY FOR DEPLOYMENT 🚀

The database is now properly set up and the application is ready to be deployed to production.

Completed By: DeepAgent
Date: October 23, 2025
Time: ~30 minutes