Vercel Build Fixes Applied

Issues Resolved

1. Database Provider Mismatch ✅

Problem: Prisma schema was configured for SQLite but .env had PostgreSQL URLs

Solution: Updated `prisma/schema.prisma` to use PostgreSQL provider with proper environment variables

```
datasource db {
    provider = "postgresql"
    url      = env("DATABASE_URL")
    directUrl = env("DIRECT_URL")
}
```

2. Husky Configuration Issues ✅

Problem: Husky trying to install git hooks in CI environment

Solution:

- Added `is-ci` package to detect CI environment
- Updated prepare script: `"prepare": "is-ci || husky install"`
- Added `HUSKY=0` environment variable to Vercel config

3. Vercel Build Configuration ✅

Problem: Build commands not optimized for Vercel environment

Solution: Updated `vercel.json` with:

- Disabled Husky in install and build commands
- Added explicit Prisma generation
- Disabled Next.js telemetry
- Added environment variables for CI

```
{
  "framework": "nextjs",
  "installCommand": "HUSKY=0 NPM_CONFIG_LEGACY_PEER_DEPS=1 npm install --no-audit --no-fund",
  "buildCommand": "HUSKY=0 npx prisma generate --schema=prisma/schema.prisma && NEXT_TELEMETRY_DISABLED=1 next build",
  "env": {
    "HUSKY": "0",
    "NEXT_TELEMETRY_DISABLED": "1"
  }
}
```

4. Environment Variables Documentation ✅

Problem: Missing comprehensive environment variable documentation

Solution:

- Updated `.env.example` with all required variables

- Created `VERCEL_DEPLOYMENT.md` with deployment guide
- Added clear documentation for required vs optional variables

5. Prisma Client Generation ✅

Problem: Prisma client not properly generated in CI
Solution:
- Updated postinstall script to explicitly specify schema path
- Added fallback error handling for DNS restrictions
- Ensured Prisma generation runs before build

Verification

✅ Local build passes: `npm run build`
✅ CI-safe build passes: `HUSKY=0 npm run build`
✅ Prisma client generates correctly
✅ All UI components are present and working
✅ TypeScript compilation successful (only ESLint warnings remain)

Next Steps

1. **Deploy to Vercel**: Push changes and redeploy
2. **Configure Environment Variables**: Set all required variables in Vercel dashboard
3. **Database Setup**: Ensure PostgreSQL database is accessible from Vercel
4. **Test Deployment**: Verify all endpoints work in production

Files Modified

- `prisma/schema.prisma` - Fixed database provider
- `vercel.json` - Optimized build configuration
- `package.json` - Added is-ci dependency and updated scripts
- `.env.example` - Comprehensive environment variables
- `VERCEL_DEPLOYMENT.md` - Deployment guide (new)

All changes have been committed and pushed to the `feature/phase1-redux` branch.