

Dependencies Notes and Recommendations

This document contains notes about package versions and potential issues found during the code audit.

Current Version Status

Potential Issues Found

1. ESLint Version Conflict

Current versions:

- eslint : 9.24.0
- eslint-config-next : 15.3.0

Issue: The ESLint version (9.x) may have compatibility issues with `eslint-config-next` which expects ESLint 8.x.

Recommendation:

```
{
  "eslint": "^8.57.0",
  "eslint-config-next": "14.2.28"
}
```

Or upgrade Next.js:

```
{
  "next": "15.3.0",
  "eslint-config-next": "15.3.0",
  "eslint": "^9.24.0"
}
```

2. UUID Package Version

Current version: `uuid@13.0.0`

Issue: Version 13.x is very new and may not be stable.

Recommendation:

```
{
  "uuid": "^10.0.0"
}
```

3. Zod Version

Current version: `zod@4.1.12`

Note: Version 4.x is relatively new. If you encounter issues, consider:

```
{  
  "zod": "^3.23.0"  
}
```

Next.js Version

Current: next@14.2.28

Latest stable: next@15.x

Note: The project uses Next.js 14.x which is stable. Upgrading to 15.x would require:

- Testing all pages and API routes
- Reviewing breaking changes
- Updating related packages

Recommendation: Stay on 14.x for now, upgrade after thorough testing.

Dependency Installation Order

First Time Setup

```
# 1. Install Node.js dependencies  
yarn install  
  
# 2. Generate Prisma clients  
npx prisma generate  
  
# 3. Install Python backend dependencies  
pip install -r backend/requirements.txt  
  
# 4. Install Python scripts dependencies (optional)  
pip install -r scripts/requirements.txt
```

After Pulling Changes

```
# Check for new dependencies  
yarn install  
  
# Regenerate Prisma client if schema changed  
npx prisma generate  
  
# Update Python dependencies if changed  
pip install -r backend/requirements.txt --upgrade
```

Key Dependencies Overview

Frontend Core

| Package | Version | Purpose |
|------------|---------|-----------------|
| next | 14.2.28 | React framework |
| react | 18.2.0 | UI library |
| typescript | 5.2.2 | Type safety |

UI Components

| Package | Version | Purpose |
|--------------|---------|------------------------|
| @radix-ui/* | Various | Headless UI components |
| tailwindcss | 3.3.3 | Utility CSS |
| lucide-react | Latest | Icons |

Forms & Validation

| Package | Version | Purpose |
|---------------------|---------|------------------------|
| react-hook-form | Latest | Form management |
| zod | 4.1.12 | Schema validation |
| @hookform/resolvers | Latest | Form + Zod integration |

Database & Backend

| Package | Version | Purpose |
|----------------|---------|-------------------|
| @prisma/client | 6.7.0 | Database ORM (JS) |
| prisma | 6.7.0 | Prisma CLI |

Python Backend

| Package | Version | Purpose |
|----------------------|---------|----------------------|
| fastapi | 0.115.0 | Python API framework |
| prisma | 0.15.0 | Prisma Python client |
| PyJWT | >=2.8.0 | JWT auth (NEW) |
| emergentintegrations | 0.1.0 | Stripe integration |

Payments

| Package | Version | Purpose |
|----------------------|---------|-----------------|
| @stripe/stripe-js | 8.2.0 | Stripe frontend |
| emergentintegrations | 0.1.0 | Stripe backend |

Storage

| Package | Version | Purpose |
|--------------------|---------|-----------------|
| @aws-sdk/client-s3 | Latest | S3 file storage |

AI/ML

| Package | Version | Purpose |
|------------------------|---------|-----------------|
| openai | Latest | OpenAI API |
| @qdrant/js-client-rest | Latest | Vector database |

TypeScript Configuration

Current Settings

The project uses strict TypeScript settings:

```
{
  "compilerOptions": {
    "strict": true,
    "noEmit": true,
    "esModuleInterop": true,
    "skipLibCheck": true
  }
}
```

Known TypeScript Issues

1. **Prisma Generated Types:** May show errors until you run `npx prisma generate`
2. **Path Aliases:** Uses `@/*` for imports, defined in `tsconfig.json`

Updating Dependencies

Safe Update Process

```
# 1. Check for outdated packages
yarn outdated

# 2. Update non-breaking changes only
yarn upgrade --scope @radix-ui
yarn upgrade --scope @hookform

# 3. Test thoroughly after updates
yarn dev
yarn build

# 4. Run type checking
yarn type-check
```

Major Version Updates

Before updating major versions:

1. **Read changelog** for breaking changes
2. **Update one package at a time**
3. **Test thoroughly**
4. **Commit after each successful update**

Recommended Update Schedule

- **Patch updates** (0.0.X): Weekly
- **Minor updates** (0.X.0): Monthly
- **Major updates** (X.0.0): Quarterly (with thorough testing)

Security Considerations

Package Audit

```
# Check for vulnerabilities
yarn audit

# Fix automatically if possible
yarn audit fix

# Check Python packages
pip-audit
```

Dependency Security

1. **Pin critical versions** in production
 2. **Use lockfile** (`yarn.lock`)
 3. **Regular security audits**
 4. **Monitor for CVEs**
-

Troubleshooting

Common Issues

“Cannot find module ‘@prisma/client’”

```
npx prisma generate
```

“Type error in Next.js build”

```
# Clear cache and rebuild  
rm -rf .next  
yarn build
```

“Module not found”

```
# Reinstall dependencies  
rm -rf node_modules yarn.lock  
yarn install
```

Python Import Errors

```
# Reinstall Python packages  
pip install -r backend/requirements.txt --force-reinstall
```

Production Considerations

Lock File Management

- **Commit** `yarn.lock` to repository
- **Never** manually edit lockfile
- **Use** `yarn install --frozen-lockfile` in CI/CD

Environment-Specific Dependencies

Some packages are only needed in development:

Development only:

- `@types/*` packages
- `eslint` and plugins
- `typescript`
- `prisma` CLI (not `@prisma/client`)

Production required:

- next
 - react
 - @prisma/client
 - Runtime dependencies
-

Notes for Future Updates

When Upgrading Next.js 14 → 15

Required changes:

1. Update all @next/* packages
2. Review [Next.js 15 migration guide](https://nextjs.org/docs/app/building-your-application/upgrading/version-15) (<https://nextjs.org/docs/app/building-your-application/upgrading/version-15>)
3. Test all API routes (may have breaking changes)
4. Update ESLint config
5. Test middleware and routing

When Upgrading React 18 → 19

Required changes:

1. Review React 19 breaking changes
 2. Test all components
 3. Update types (@types/react)
 4. Check third-party component compatibility
-

Recommended Actions

Immediate (Before Production Deploy)

- [] Consider downgrading ESLint to 8.x OR upgrading Next.js to 15.x
- [] Downgrade uuid to 10.x
- [] Run yarn audit and fix vulnerabilities
- [] Test all critical paths

Short-term (Within 1-2 Weeks)

- [] Set up automated dependency updates (Dependabot/Renovate)
- [] Add yarn audit to CI/CD pipeline
- [] Document any custom package patches

Long-term (Next Quarter)

- [] Plan Next.js 15 upgrade
 - [] Review and update all major dependencies
 - [] Add integration tests for critical features
-

Last Updated: December 11, 2025