Memento Mori - Deployment Guide

📋 Overview

This guide provides step-by-step instructions for deploying the Memento Mori mortality calculator application to Vercel or Netlify. The application is a Next.js 15 app with TypeScript and Tailwind CSS.

🚀 Quick Deploy Options

Option 1: Vercel Deployment (Recommended for Next.js)

Prerequisites

- GitHub, GitLab, or Bitbucket account
- Vercel account (free tier available)

Method A: Deploy from Git Repository

1. **Upload to Git Repository**
   - Create a new repository on GitHub/GitLab/Bitbucket
   - Upload all files from this package to the repository
   - Ensure the `app/` directory is the root of your repository

2. **Deploy on Vercel**
   - Go to vercel.com (https://vercel.com) and sign in
   - Click “New Project”
   - Import your Git repository
   - Vercel will auto-detect it’s a Next.js project
   - **Important**: Set the root directory to `app/` if your repository structure includes the outer folder
   - Configure build settings:
     - **Build Command**: `npm run build`
     - **Output Directory**: `.next` (auto-detected)
     - **Install Command**: `npm install`
     - Click “Deploy”

Method B: Deploy via Vercel CLI

1. **Install Vercel CLI**

```bash
npm i -g vercel
```

2. **Navigate to app directory and deploy**

```bash
cd app/
vercel
```

   - Follow the prompts
   - Choose “yes” when asked if you want to deploy
   - Select your scope/team
   - Link to existing project or create new one

Option 2: Netlify Deployment

Method A: Drag & Drop Deployment

1. **Build the application locally**
   ```
   bash
   cd app/
   npm install
   npm run build
   ```

2. **Deploy to Netlify**
   - Go to netlify.com (https://netlify.com) and sign in
   - Drag and drop the `app/.next` folder to the deploy area
   - **Note**: This method requires manual updates for each deployment

Method B: Git-based Deployment

1. **Upload to Git Repository**
   - Create a new repository on GitHub/GitLab/Bitbucket
   - Upload all files from this package
   - Ensure the `app/` directory structure is maintained

2. **Deploy on Netlify**
   - Go to netlify.com (https://netlify.com) and sign in
   - Click “New site from Git”
   - Connect your Git provider and select the repository
   - Configure build settings:
     - **Base directory**: `app/`
     - **Build command**: `npm run build`
     - **Publish directory**: `app/.next`
     - Click “Deploy site”

Method C: Netlify CLI

1. **Install Netlify CLI**
   ```
   bash
   npm install -g netlify-cli
   ```

2. **Build and deploy**
   ```
   bash
   cd app/
   npm install
   npm run build
   netlify deploy --prod --dir=.next
   ```

⚙️ Configuration Details

Build Settings Summary

- **Framework**: Next.js
- **Node.js Version**: 18.x or higher
- **Build Command**: `npm run build`
- **Output Directory**: `.next`
- **Install Command**: `npm install`

Environment Variables

This application doesn't require any environment variables for basic functionality. All calculations are performed client-side.

Dependencies

The application uses the following key dependencies:
- Next.js 15.4.6
- React 19.1.0
- TypeScript 5.x
- Tailwind CSS 4.x

🔧 Troubleshooting

Common Issues and Solutions

1. Build Fails with "Module not found"

Solution: Ensure all dependencies are installed

```
cd app/
rm -rf node_modules package-lock.json
npm install
npm run build
```

2. Vercel: "No Output Directory"

Solution: Ensure the root directory is set correctly
- In Vercel dashboard, go to Project Settings → General
- Set "Root Directory" to `app/` if needed

3. Netlify: "Build Failed"

Solution: Check build settings
- Base directory: `app/`
- Build command: `npm run build`
- Publish directory: `app/.next`

4. TypeScript Errors During Build

Solution: The project is configured with strict TypeScript. If you encounter type errors:

```
cd app/
npm run lint
```

Fix any reported issues before deploying.

5. Tailwind CSS Not Loading

Solution: Ensure PostCSS configuration is correct
- Check that `postcss.config.mjs` exists in the app directory
- Verify `globals.css` imports Tailwind directives

6. Large Bundle Size Warning

Solution: This is normal for the mortality calculator with all its statistical data. The app is optimized for performance with:

- Client-side rendering for calculations
- Efficient component structure
- Minimal external dependencies

Performance Optimization

The application is already optimized with:
- Next.js automatic code splitting
- Client-side calculations (no server load)
- Responsive design for all devices
- Local storage for scenario persistence

📱 Post-Deployment Verification

After deployment, verify the following features work correctly:

1. **Homepage loads** with mortality calculator form
2. **Form submission** calculates and displays results
3. **Scenario saving** works (check browser localStorage)
4. **Navigation** between pages (About, Compare, Resources)
5. **Responsive design** on mobile devices
6. **Compare scenarios** functionality
7. **Resources page** loads with mental health resources

🔗 Useful Links

- Next.js Deployment Documentation (https://nextjs.org/docs/deployment)
- Vercel Documentation (https://vercel.com/docs)
- Netlify Documentation (https://docs.netlify.com/)

📞 Support

If you encounter issues not covered in this guide:
1. Check the build logs in your deployment platform
2. Verify all files are uploaded correctly
3. Ensure Node.js version compatibility (18.x+)
4. Review the troubleshooting section above

Estimated Deployment Time: 5-10 minutes
Recommended Platform: Vercel (optimized for Next.js)