Provide the instructions on how to deploy this to Vercel or netlify. Indicate if this can be done using CLI

Based on the TETRIX website specifications and the comprehensive documentation provided, here are the detailed deployment instructions for both Vercel and Netlify platforms, including CLI deployment options.

Vercel Deployment

Prerequisites

- Ensure your Astro project is built and tested locally
- Have a Git repository (GitHub, GitLab, or Bitbucket) with your code
- Node.js version 18.14.0 or later installed

Method 1: Vercel Website UI Deployment

Step 1: Prepare Your Project

```
# Build your project locally to test
npm run build
```

Step 2: Deploy via Vercel Dashboard

1. Push your TETRIX code to your Git repository
2. Visit vercel.com and sign in
3. Click "Add New Project" or "Import Project"
4. Select your Git provider and authorize Vercel
5. Choose your TETRIX repository
6. Vercel will automatically detect Astro and configure:
    - **Build Command**: `astro build` or `npm run build`
    - **Output Directory**: `dist`
    - **Install Command**: `npm install`

Step 3: Deploy

- Click "Deploy" - Vercel handles the rest automatically
- Your site will be live at a URL like `tetrix-project.vercel.app`

- Future Git pushes will trigger automatic deployments

Method 2: Vercel CLI Deployment

Step 1: Install Vercel CLI

```
# Install globally
npm install -g vercel

# Or install locally for CI environments
npm install vercel --save-dev
```

Step 2: Login and Initialize

```
# Login to Vercel
vercel login

# Navigate to your project directory
cd your-tetrix-project

# Deploy (first time setup)
vercel
```

Step 3: Follow CLI Prompts

When running `vercel` for the first time:

- **Set up and deploy?** → Yes
- **Which scope?** → Select your account/team
- **Link to existing project?** → No (for new deployment)
- **Project name?** → `tetrix-website` (or your preferred name)
- **Directory with code?** → `./` (current directory)
- **Override settings?** → No (Vercel auto-detects Astro)

Step 4: Production Deployment

```
# For production deployment
vercel --prod

# For preview deployment (default)
vercel
```

Advanced Vercel Configuration

For server-side rendering or additional Vercel features, add the Vercel adapter:

```
# Add Vercel adapter
npm install @astrojs/vercel
```

Update your `astro.config.mjs`:

```
import { defineConfig } from 'astro/config';
import vercel from '@astrojs/vercel/static';

export default defineConfig({
  output: 'static', // or 'server' for SSR
  adapter: vercel({
    webAnalytics: { enabled: true },
    maxDuration: 8,
  }),
});
```

Netlify Deployment

Method 1: Netlify Website UI Deployment

Step 1: Prepare Repository

- Ensure your TETRIX project is pushed to Git (GitHub, GitLab, Bitbucket, or Azure DevOps)

Step 2: Deploy via Netlify Dashboard

1. Visit netlify.com and sign in
2. Click "Add new site" → "Import an existing project"
3. Choose your Git provider and authorize
4. Select your TETRIX repository
5. Netlify auto-detects Astro settings:
    - **Build command**: `astro build` or `npm run build`
    - **Publish directory**: `dist`
    - **Node.js version**: 18.20.8 or higher

Step 3: Deploy

- Click "Deploy site"
- Your site goes live at a URL like `amazing-tetrix-123456.netlify.app`
- Automatic deployments on Git pushes

Method 2: Netlify CLI Deployment

Step 1: Install Netlify CLI

```
# Install globally
npm install -g netlify-cli

# Or locally for CI
npm install netlify-cli --save-dev
```

Step 2: Login and Initialize

```
# Login to Netlify
netlify login

# Navigate to project directory
cd your-tetrix-project

# Initialize new site
netlify init
```

Step 3: Follow CLI Setup

- **Create & configure a new site** → Yes
- **Team** → Select your team
- **Site name** → `tetrix-website`
- **Build command** → `astro build` (auto-detected)
- **Directory to deploy** → `dist` (auto-detected)
- **Netlify functions directory** → (leave empty unless using functions)

Step 4: Deploy Options

```
# Deploy to preview URL
netlify deploy

# Deploy to production
netlify deploy --prod

# Deploy specific directory
netlify deploy --dir=dist --prod
```

Manual Netlify Deployment

For CI/CD environments or manual control:

```
# Build locally
npm run build

# Deploy the build
netlify deploy --dir=dist --prod
```

Configuration Files

Optional: netlify.toml

Create `netlify.toml` in your project root for advanced configuration:

```
[build]
  command = "npm run build"
  publish = "dist"

[build.environment]
  NODE_VERSION = "18.20.8"

[[redirects]]
  from = "/*"
  to = "/index.html"
  status = 200
```

Optional: vercel.json

Create `vercel.json` for Vercel-specific configurations:

```
{
  "buildCommand": "npm run build",
  "outputDirectory": "dist",
  "framework": "astro",
  "rewrites": [
    {
      "source": "/(.*)",
      "destination": "/index.html"
    }
  ]
}
```

Deployment Checklist

Pre-deployment:

- [ ] Test build locally: `npm run build`
- [ ] Verify all environment variables are set
- [ ] Ensure Firebase/Firestore credentials are configured
- [ ] Test all routes and functionality
- [ ] Optimize images and assets

Post-deployment:

- [ ] Test live site functionality
- [ ] Verify contact forms work
- [ ] Check authentication flows
- [ ] Test responsive design on various devices

- [ ] Set up custom domain (optional)
- [ ] Configure SSL certificate (automatic on both platforms)

CI/CD Integration

Both platforms support automatic deployments:

Vercel: Automatically deploys on Git pushes to main branch
Netlify: Supports build hooks, scheduled builds, and Git-triggered deployments

For the TETRIX project with Firebase authentication, ensure your environment variables are properly configured in both platforms' dashboard settings before deployment.

Both Vercel and Netlify offer excellent support for Astro projects with zero-configuration deployment, making them ideal choices for the TETRIX SaaS website.

⁂