Aura Design Studio

📋 Table of Contents

🎯 Overview

Aura Design Studio is a modern web application built with cutting-edge technologies and powered by Lovable, an AI-powered platform for rapid application development. This project showcases a clean, type-safe architecture with a focus on developer experience and performance.

Repository: https://github.com/johnwesly08/aura-design-studio

🛠 Technology Stack

This project leverages a modern and robust technology stack optimized for performance and developer productivity:

Core Technologies

- **Vite -** Lightning-fast build tool and development server with Hot Module Replacement (HMR)
- **TypeScript** (95.8% of codebase) **-** Strongly-typed JavaScript for enhanced code quality and IDE support
- **React -** Declarative UI library for building component-based user interfaces

UI & Styling

- **shadcn-ui -** Premium, accessible component library built on Radix UI
- **Tailwind CSS** (3.3% of codebase) **-** Utility-first CSS framework for rapid, responsive design

Build & Development Tools

- **Bun -** Fast all-in-one JavaScript runtime and package manager
- **ESLint -** JavaScript/TypeScript linter for code quality
- **PostCSS -** CSS transformation and processing tool
- **Vitest -** Fast unit testing framework powered by Vite

Development Platform

- **Lovable -** AI-powered development platform for rapid prototyping and deployment

📁 Project Structure

```
aura-design-studio/
├── public/            # Static assets (images, fonts, etc.)
├── src/               # Source code
│   ├── components/     # Reusable React components
│   │   └── ui/        # shadcn-ui components
│   ├── pages/         # Page-level components
│   ├── hooks/         # Custom React hooks
│   ├── lib/           # Utility functions and helpers
│   ├── types/         # TypeScript type definitions
│   ├── styles/        # Global styles and CSS
│   └── App.tsx        # Root application component
├── .gitignore         # Git ignore rules
├── components.json     # shadcn-ui configuration
├── eslint.config.js    # ESLint configuration
├── index.html         # HTML entry point
├── package.json       # Dependencies and scripts
├── postcss.config.js   # PostCSS configuration
├── tailwind.config.ts  # Tailwind CSS configuration
├── tsconfig.json       # TypeScript configuration
├── tsconfig.app.json   # TypeScript app-specific config
├── tsconfig.node.json  # TypeScript Node.js config
├── vite.config.ts      # Vite configuration
└── vitest.config.ts    # Vitest testing configuration
```

🚀 Getting Started

Prerequisites

Before you begin, ensure you have the following installed:

- **Node.js** (v18 or higher recommended)
  - Install via nvm for easy version management
- **npm** (comes bundled with Node.js)

Installation

1. **Clone the repository**

```bash
git clone https://github.com/johnwesly08/aura-design-studio.git
```

2. **Navigate to the project directory**

```bash
cd aura-design-studio
```

3. **Install dependencies**

```bash
npm install
```

4. **Start the development server**

```bash
npm run dev
```

The application will launch and be available at `http://localhost:5173` (or another port if 5173 is in use).

💻 Development

Available Scripts

```bash
# Start development server with hot-reload
npm run dev

# Build for production
npm run build

# Preview production build locally
npm run preview

# Run linter to check code quality
npm run lint

# Run tests
npm run test

# Run tests in watch mode
npm run test:watch

# Run tests with coverage
npm run test:coverage
```

Development Workflow Options

1. Using Lovable Platform (Recommended)

The fastest way to develop with AI assistance:

1. Visit your Lovable Project
2. Use natural language prompts to make changes
3. Changes are automatically committed to this repository
4. Preview changes instantly in the Lovable interface

Benefits:

- AI-powered development
- Instant previews
- Automatic Git commits
- No local setup required

2. Using Your Local IDE

For traditional development workflow:

1. Clone the repository (see Installation)
2. Make changes in your preferred editor (VS Code, WebStorm, etc.)
3. Test locally with `npm run dev`
4. Commit and push changes
5. Changes sync automatically with Lovable

Recommended VS Code Extensions:

- ESLint
- Prettier
- Tailwind CSS IntelliSense
- TypeScript and JavaScript Language Features

3. Using GitHub Web Editor

Quick edits without local setup:

1. Navigate to any file in the repository
2. Click the **Edit** button (pencil icon)
3. Make your changes in the web editor
4. Commit directly to the main branch or create a new branch

4. Using GitHub Codespaces

Full cloud-based development environment:

1. Click the **Code** button on the repository page
2. Select the **Codespaces** tab
3. Click **New codespace**
4. Develop in a fully-configured VS Code environment in your browser

Code Style and Quality

This project maintains high code quality standards:

- **TypeScript** for type safety (95.8% of codebase)

- **ESLint** for code linting and consistency
- **Prettier** (recommended) for code formatting

Before committing, ensure:

```bash
# Check for linting errors
npm run lint

# Run tests
npm run test

# Build successfully
npm run build
```

Adding New Components

To add a new shadcn-ui component:

```bash
npx shadcn-ui@latest add [component-name]
```

Example:

```bash
npx shadcn-ui@latest add button
npx shadcn-ui@latest add card
npx shadcn-ui@latest add dialog
```

🌐 Deployment

Deploy with Lovable (One-Click)

The simplest deployment method:

1. Open your Lovable Project
2. Navigate to **Share** → **Publish**
3. Your app is now live with a public URL
4. Share your project instantly

Features:

- Automatic SSL certificates
- Global CDN distribution
- Instant deployments
- Zero configuration

Custom Domain Setup

Connect your own domain:

1. In Lovable, go to **Project** → **Settings** → **Domains**
2. Click **Connect Domain**
3. Enter your domain name
4. Configure DNS records as instructed
5. SSL certificate is provisioned automatically

📚 **Detailed Guide:** Custom Domain Documentation

Alternative Deployment Options

Vercel

bash

```
npm install -g vercel
vercel
```

Netlify

bash

```
npm install -g netlify-cli
netlify deploy
```

Manual Build

bash

```
npm run build
# Deploy the 'dist' folder to your hosting provider
```

⚙️ Configuration

TypeScript Configuration

The project uses three TypeScript configurations:

- **`tsconfig.json`** - Base configuration
- **`tsconfig.app.json`** - Application-specific settings
- **`tsconfig.node.json`** - Node.js/build tool settings

Tailwind CSS Customization

Modify `tailwind.config.ts` to customize:

```typescript
export default {
  theme: {
    extend: {
      colors: {
        // Add custom colors
      },
      fontFamily: {
        // Add custom fonts
      },
      spacing: {
        // Add custom spacing
      },
    },
  },
}
```

shadcn-ui Configuration

Component settings in `components.json`:

```json
```

```
{
  "style": "default",
  "rsc": false,
  "tsx": true,
  "tailwind": {
    "config": "tailwind.config.ts",
    "css": "src/index.css",
    "baseColor": "slate",
    "cssVariables": true
  },
  "aliases": {
    "components": "@/components",
    "utils": "@/lib/utils"
  }
}
```

Vite Configuration

Build and development settings in `vite.config.ts`:

- Path aliases
- Build optimizations
- Development server settings
- Plugin configurations

🎨 Features

Based on the project structure and technologies, Aura Design Studio includes:

Core Features

- ⚡ **Lightning-Fast Performance -** Powered by Vite for instant HMR
- 🎯 **Type-Safe Development -** Full TypeScript support (95.8% coverage)
- 🎨 **Modern UI Components -** Pre-built shadcn-ui components
- 📱 **Responsive Design -** Mobile-first approach with Tailwind CSS
- 🧪 **Testing Ready -** Vitest configuration for unit and integration tests
- 🔄 **Hot Module Replacement -** Instant feedback during development
- 📦 **Optimized Builds -** Production-ready build optimization

Developer Experience

- 🤖 **AI-Powered Development -** Lovable platform integration
- 🛠️ **Modern Tooling -** ESLint, PostCSS, TypeScript
- 📝 **Component Library -** Extensible shadcn-ui components
- 🎯 **Path Aliases -** Clean imports with `@/` prefix
- 🔍 **IntelliSense Support -** Full TypeScript autocomplete

🤝 Contributing

Contributions are welcome! Here's how you can help:

Getting Started

1. **Fork the repository**
2. **Create a feature branch**

```bash
git checkout -b feature/amazing-feature
```

3. **Make your changes**
4. **Commit your changes**

```bash
git commit -m 'Add some amazing feature'
```

5. **Push to the branch**

```bash
git push origin feature/amazing-feature
```

6. **Open a Pull Request**

Contribution Guidelines

- ✅ Follow the existing code style and conventions
- ✅ Write meaningful commit messages (use conventional commits)
- ✅ Add tests for new features

- ✅ Update documentation as needed
- ✅ Ensure all tests pass before submitting
- ✅ Keep pull requests focused and atomic

Commit Message Format

```
<type>(<scope>): <subject>

<body>

<footer>
```

Types: feat, fix, docs, style, refactor, test, chore

Example:

```
feat(components): add new Card component

Added a reusable Card component with variants for different use cases.

Closes #123
```

📞 Support

Getting Help

- 📖 **Documentation**: Check this README and inline code comments
- 🐛 **Bug Reports**: GitHub Issues
- 💡 **Feature Requests**: GitHub Issues
- 📚 **Lovable Docs**: https://docs.lovable.dev

Troubleshooting

Build Errors

Problem: Build fails with dependency errors

```
bash
```

```
# Solution: Clean install
rm -rf node_modules package-lock.json
npm install
```

Problem: TypeScript errors

```bash
# Solution: Check TypeScript configuration
npm run type-check
```

Development Server Issues

Problem: Port already in use

```bash
# Solution: Use a different port
npm run dev -- --port 3000
```

Problem: Changes not reflecting

```bash
# Solution: Clear Vite cache
rm -rf node_modules/.vite
npm run dev
```

Performance Issues

Problem: Slow build times

```bash
# Solution: Clear cache and rebuild
rm -rf node_modules/.vite dist
npm run build
```

Common Questions

Q: How do I add a new page? A: Create a new component in `src/pages/` and add routing in your router configuration.

Q: How do I customize the theme? A: Modify `tailwind.config.ts` for Tailwind customization and `components.json` for shadcn-ui theming.

Q: Can I use this project without Lovable? A: Yes! This is a standard Vite + React + TypeScript project that works independently.

Q: How do I deploy to my own server? A: Run `npm run build` and deploy the `dist` folder to any static hosting service.

📄 License

This project's license information is not specified. Please contact the repository owner for licensing details.

🙏 Acknowledgments

- Built with Lovable
- UI components from shadcn-ui
- Styled with Tailwind CSS
- Powered by Vite

Built with ❤️ using Lovable

Last Updated: January 2026