

Delly Portfolio Documentation

Tech Stack

- Next.js 14
- React 18
- TypeScript
- Tailwind CSS
- Framer Motion
- React Icons
- React Hot Toast
- React Intersection Observer
- React Scroll
- React Spring
- React Three Fiber
- Three.js
- Drei
- GSAP
- EmailJS
- React Email
- Resend
- Nodemailer
- Express.js
- MongoDB
- Mongoose
- JWT
- Bcrypt
- Cors
- Dotenv
- Nodemon
- Concurrently
- Markdown PDF

Project Structure

- client/ - Frontend Next.js application
 - app/ - Next.js app directory
 - api/ - API routes
 - (routes)/ - Page routes
 - components/ - Reusable components
 - lib/ - Utility functions
 - styles/ - Global styles
 - public/ - Static assets
 - scripts/ - Utility scripts
- server/ - Backend Express.js application

- config/ - Configuration files
- controllers/ - Route controllers
- middleware/ - Custom middleware
- models/ - Database models
- routes/ - API routes
- utils/ - Utility functions

Key Features

1. Modern UI/UX

- Responsive design using Tailwind CSS
- Smooth animations with Framer Motion and GSAP
- 3D elements with React Three Fiber
- Dark/Light mode support
- Custom cursor effects
- Loading states and transitions

2. Performance Optimization

- Image optimization with Next.js
- Code splitting and lazy loading
- Server-side rendering
- Static site generation
- Caching strategies
- Bundle size optimization

3. Contact System

- Email integration with EmailJS
- Form validation
- Success/error notifications
- Spam protection
- Backup email system with Nodemailer

4. Authentication

- JWT-based authentication
- Secure password hashing with bcrypt
- Protected routes
- Session management
- Role-based access control

5. Database

- MongoDB with Mongoose
- Data modeling
- CRUD operations
- Data validation
- Error handling

6. API

- RESTful API design
- Error handling
- Rate limiting
- CORS configuration
- Request validation

Implementation Details

Frontend Architecture

- Component-based architecture
- Custom hooks for reusable logic
- Context API for state management
- Error boundaries for graceful error handling
- Loading states for better UX
- Responsive design patterns
- Accessibility features

Backend Architecture

- MVC pattern
- Middleware for common functionality
- Error handling middleware
- Request validation
- Security best practices
- Database optimization
- API documentation

Security Features

- JWT authentication
- Password hashing
- CORS protection
- Rate limiting
- Input validation
- XSS protection
- CSRF protection

Performance Features

- Code splitting
- Lazy loading
- Image optimization
- Caching strategies
- Bundle optimization
- Server-side rendering
- Static generation

Development Workflow

1. Feature planning
2. Development

3. Testing
4. Code review
5. Deployment
6. Monitoring

Deployment

- Vercel for frontend
- Railway for backend
- MongoDB Atlas for database
- Environment variables management
- CI/CD pipeline
- Monitoring and logging

Maintenance

- Regular updates
- Security patches
- Performance monitoring
- Error tracking
- User feedback
- Analytics