

Najahi – Personal Portfolio Resume Template

Developed by Kraitto

Version 1.0.0

December 28, 2024

Welcome

Thank you for purchasing Najahi, our Next.js-based personal portfolio resume template. This comprehensive documentation covers the system overview, features, and detailed installation procedures.

Support Channels

For any inquiries beyond the scope of this documentation, we're here to help:

Technical Support: **support@kraitto.com**

Website: **www.kraitto.com**

Contact Form: Available on our website

Social Media: Follow us for updates and support

Tutorials: **<https://github.com/kraittoofficial/najahi-preview/>**

Package Contents

Core Package (najahi-v1.0.0.zip)

.env.example - Environment configuration template

Dockerfile - Container configuration file

docker-compose.yml - Docker orchestration configuration

Ecosystem.config.js - PM2 process management settings

LICENSE.txt - Software license agreement

Documentation

Najahi v1.0.0 Documentation.pdf - Comprehensive user and deployment guide

Preview Images - Visual reference materials

Additional Support

For general inquiries and discussions:

Visit our Envato Market "Item Discussion" section

Review our documentation

Contact our support team

We're committed to providing excellent support for your Najahi implementation. While we can't guarantee immediate solutions to all issues, we strive to assist you promptly and effectively.

Mohammad Saiful Islam

Founder & Partner

Kraitto

Table of Contents

# Najahi Introduction.....	4
# Key Features & Benefits.....	4
Core Features.....	4
Portfolio Components.....	4
Professional Features.....	4
# Technical Implementation.....	4
Core Technology Stack.....	4
Frontend Framework:.....	4
Styling & UI:.....	4
Development Tools.....	5
Code Quality:.....	5
Build & Deployment:.....	5
API Integration.....	5
LinkedIn API:.....	5
Contact Form:.....	5
# Customization.....	5
# System Requirements.....	5
Development Environment.....	5
Production Environment.....	5
# Getting Started.....	5
# Support.....	5
# License.....	6
# Portfolio Customization.....	7
# Header Customization (header.tsx).....	7
# Hero Section (hero.tsx).....	7
# About Section (about.tsx).....	7
# Experience Section (experience.tsx).....	7
# Education Section (education.tsx).....	7
# Skills Section (skills.tsx).....	8
# Core Expertise Section (core-expertise.tsx).....	8
# Projects Section (projects.tsx).....	8
# LinkedIn Posts Section (linkedin-posts.tsx).....	8
# Contact Section (contact.tsx).....	8
# Footer Section (footer.tsx).....	8
# Theme Customization.....	8
Customize Global Styles (globals.css).....	8
# Analytics Setup (Analytics.tsx).....	9
Required Environment Variables.....	9
Contact Form Configuration.....	9

Security Settings.....	9
# SEO Customization.....	9
Update Metadata Base (layout.tsx).....	9
Page-Specific SEO (page.tsx).....	10
Important Notes for SEO Customization.....	10
# Important Notes for Portfolio Customization.....	11
# Initial Setup.....	12
1. System Requirements Verification.....	12
2. Email Server Configuration.....	12
# Deployment Methods.....	13
Method 1: Vercel Deployment.....	13
Pre-deployment Setup.....	13
Deployment Steps.....	13
Method 2: Docker Deployment.....	14
Pre-deployment Setup.....	14
Configuration Files.....	14
Deployment Steps.....	16
Method 3: PM2 Deployment.....	17
Installation & Setup.....	17
Configuration.....	17
Deployment Steps.....	17
# Troubleshooting Guide.....	18
# Security Checklist.....	18

Najahi | Personal Portfolio Resume Template

Najahi Introduction

Najahi is a personal portfolio resume template built with Next.js 15 and React 19. It includes a responsive layout for various devices, dual themes with light and dark modes, and integrated SEO features for enhanced visibility. The template provides features like a customizable hero section, work timeline, skill matrices, and a protected contact system to create an organized online presence.

Key Features & Benefits

Core Features

- Next-Generation Architecture: Built with Next.js 15 and React 19 for optimal performance
- Responsive Design: Fluid layouts that adapt seamlessly across all devices (mobile, tablet, desktop)
- Dual Theme Support: Integrated light and dark mode with smooth transitions
- SEO Optimized: Built-in SEO best practices and meta tag optimization

Portfolio Components

- Dynamic Hero Section: Customizable introduction with animated components
- Interactive Work Timeline: Engaging presentation of professional experience
- Skills Visualization: Modern progress indicators and skill matrices
- Project Showcase: Grid-based portfolio with filtering capabilities
- LinkedIn Integration: Real-time LinkedIn post embedding with automatic updates
- Protected Contact System: Turnstile-secured contact form with spam prevention

Professional Features

- Analytics Ready: Built-in support for Google Analytics and custom tracking
- Automated Deployment: Streamlined deployment process with PM2 integration
- Image Optimization: Next.js Image component for optimal loading
- Form Handling: Server-side validation and email notification system
- Custom API Routes: Dedicated endpoints for LinkedIn posts and contact form

Technical Implementation

Core Technology Stack

Frontend Framework:

- Next.js 15 (App Router)
- React 19 with Server Components
- TypeScript for type safety

Styling & UI:

- TailwindCSS for utility-first styling
- Shadcn UI for consistent component design
- Framer Motion for smooth animations

Development Tools

Code Quality:

- ESLint for code linting
- Prettier for code formatting
- TypeScript for static typing

Build & Deployment:

- PM2 for process management
- Docker support for containerization
- Environmental variable management

API Integration

LinkedIn API:

- Custom endpoint for fetching posts
- Caching mechanism for optimal performance
- Error handling and fallback states

Contact Form:

- Server-side validation
- Email service integration
- Rate limiting and spam protection

Customization

Najahi is designed for easy customization. Modify the content, styling, and components to align with your personal brand. See the **Customization Guide** for detailed instructions.

System Requirements

Development Environment

- Node.js ($\geq 20.x.x$)
- NPM ($\geq 10.x.x$)
- Git

Production Environment

- Minimum 4GB RAM
- 20GB storage
- Modern web browser support
- SSL certification

Getting Started

For detailed deployment instructions, please refer to the **Installation Guide**.

For tutorials, previews, and additional information, visit our **GitHub Preview Repository**.

Support

For technical support, inquiries, or customization requests, please contact us at **support@kraitto.com**.

License

This software is released under a Commercial License. For licensing details, please refer to the **LICENSE.txt** file in the root directory.

For licensing inquiries, customization requests, or support, please contact us directly.

Najahi | Customization Guide

Portfolio Customization

This guide will help you customize your Najahi portfolio template. Follow the sections below to personalize your portfolio.

Header Customization (header.tsx)

- Replace logo images in the /public/logo/ directory:
 - najahi-dark.png - Logo for dark theme
 - najahi-light.png - Logo for light theme
- Recommended logo dimensions: 120x40px

Hero Section (hero.tsx)

- Replace profile image:
 - Add your photo to /public/Photo.png
 - Recommended size: 200x200px, square format
- Update personal details:
 - Name: Replace "Mohammad Saiful Islam"
 - Designation: Replace "Founder & Partner"
- Update social links:
 - LinkedIn: Update linkedin.com/in/yourusername
 - GitHub: Update github.com/yourusername
 - Email: Update your.email@example.com
- Replace resume:
 - Add your resume to /public/Resume.pdf

About Section (about.tsx)

- Replace the personal profile statement in the <motion.p> component
- Remove or modify the demo content note

Experience Section (experience.tsx)

- Modify the experiences array:

```
{
  title: "Your Job Title",
  company: "Company Name",
  duration: "Start - End Date",
  responsibilities: [
    "Responsibility 1",
    "Responsibility 2",
    "Responsibility 3"
  ]
}
```

Education Section (education.tsx)

- Update educations array with your educational background
- Update certifications array with your certifications
- Follow the existing format:

```
{
  degree: "Your Degree",
  subject: "Your Subject",
  cpga: "Your CGPA",
  institution: "Institution Name",
  year: "Completion Year"
}
```

Skills Section (skills.tsx)

- Modify the skills array with your technical skills
- Add or remove skills as needed

Core Expertise Section (core-expertise.tsx)

- Update expertiseAreas array with your areas of expertise
- For each area, include:
 - Title
 - Description
 - Key achievements

Projects Section (projects.tsx)

- Update projects array with your notable projects
- For each project, include:
 - Title
 - Description
 - Technologies used
 - Project link

LinkedIn Posts Section (linkedin-posts.tsx)

- Configure LinkedIn API integration in /api/linkedin-posts
- Update LinkedIn username in the API configuration

Contact Section (contact.tsx)

- Update contact information:
 - Email address
 - Phone number
 - LinkedIn profile
 - GitHub profile
- Configure Cloudflare Turnstile:
 - Add NEXT_PUBLIC_CLOUDFLARE_TURNSTILE_SITE_KEY to your environment variables

Footer Section (footer.tsx)

- Update social media links
- Modify copyright text if needed

Theme Customization

Customize Global Styles (globals.css)

Light Theme Colors (ShadCN UI)


```
:root {
  --background: 0 0% 100%;
  --foreground: 222.2 84% 4.9%;
  --primary: 221.2 83.2% 53.3%; /* Customize your primary brand color */
  --secondary: 210 40% 96.1%; /* Customize your secondary color */
  --accent: 210 40% 96.1%; /* Customize your accent color */
  /* Modify other color variables as needed */
}
```

Dark Theme Colors (ShadCN UI)

```
.dark {
  --background: 222.2 84% 4.9%;
  --foreground: 210 40% 98%;
  --primary: 217.2 91.2% 59.8%; /* Customize dark theme primary color */
  --secondary: 217.2 32.6% 17.5%; /* Customize dark theme secondary */
  /* Modify other dark theme colors as needed */
}
```

Analytics Setup (Analytics.tsx)

Required Environment Variables

Create .env file from .env.example:

Google Analytics

NEXT_PUBLIC_GA_ID=G-XXXXXXXXXX

Microsoft Bing

NEXT_PUBLIC_BING_ID=XXXXXXXXXX

Yandex Metrika

NEXT_PUBLIC_YANDEX_ID=XXXXXXXXXX

Contact Form Configuration

Email Settings

EMAIL_USER=your@email.com

EMAIL_PASS=your-email-password

EMAIL_TO=recipient@email.com

Cloudflare Turnstile

NEXT_PUBLIC_CLOUDFLARE_TURNSTILE_SITE_KEY=your-site-key

CLOUDFLARE_TURNSTILE_SECRET_KEY=your-secret-key

Security Settings

CORS Configuration

CORS_ORIGINS=https://yourdomain.com,https://api.yourdomain.com

NODE_ENV=production

SEO Customization

Update Metadata Base (layout.tsx)

```
export const metadata: Metadata = {
```

```

metadataBase: new URL("https://yourdomain.com"), // Your actual domain
title: {
  default: "Your Name - Portfolio",
  template: "%s | Your Professional Title",
},
description: "Your professional summary in 150-160 characters",
keywords: ["your", "relevant", "keywords"],
authors: [{ name: "Your Name" }],
creator: "Your Name/Company",
};
OpenGraph Settings (layout.tsx)
openGraph: {
  type: "website",
  url: "https://yourdomain.com",
  siteName: "Your Name - Portfolio",
  images: [{
    url: "https://yourdomain.com/og-image.jpg",
    width: 1200,
    height: 630,
    alt: "Your Portfolio Preview"
  }]
}
Twitter Cards (layout.tsx)
twitter: {
  card: "summary_large_image",
  creator: "@yourhandle",
  images: ["https://yourdomain.com/twitter-image.jpg"]
}

```

Page-Specific SEO (page.tsx)

```

export const metadata: Metadata = {
  title: "Your Name - Professional Title",
  description: "Your professional summary focused on your expertise",
  openGraph: {
    title: "Your Name - Professional Title",
    description: "Compelling description for social sharing",
  },
  twitter: {
    title: "Your Name - Professional Title",
    description: "Engaging description for Twitter shares",
  },
};

```

Important Notes for SEO Customization

- All image URLs should be absolute paths
- Test metadata using social media debugging tools
- Verify analytics tracking is working
- Ensure environment variables are set in production

- Optimize images for social sharing
- Test contact form functionality
- Monitor CORS settings for API endpoints

Important Notes for Portfolio Customization

- Optimize image sizes for better performance.
- Test all links before deployment.
- Ensure personal information is accurate.
- Update and secure environment variables in the deployment platform.
- Test contact form functionality after customization.
- Replace placeholder content with actual information.
- Test the app across devices and browsers.
- Validate structured data using Google's testing tool.
- Monitor analytics after deployment.

Najahi | Installation Guide

Initial Setup

1. System Requirements Verification

Check Node.js version:

node --version # Must be v20 or later

Verify package manager installation:

npm --version # or

pnpm --version

2. Email Server Configuration

Gmail Setup (Recommended for starting):

- Enable 2-Factor Authentication
- Generate App Password
- Use following settings:
 - EMAIL_USER=
 - EMAIL_PASS=
 - EMAIL_TO=

Deployment Methods

Method 1: Vercel Deployment

Pre-deployment Setup

1. Install Vercel CLI:
`npm i -g vercel`
2. Create .env file with all required variables
3. Verify project structure follows Vercel requirements

Deployment Steps

1. Login to Vercel:
`vercel login`
2. Initial deployment:
`vercel`
3. Configure environment variables:
Open Vercel dashboard
Go to Project Settings
Navigate to Environment Variables
Import .env file or add manually
Verify all required variables are set
4. Production deployment:
`vercel --prod`

Method 2: Docker Deployment

Pre-deployment Setup

1. Install Docker:

```
# Ubuntu
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh

# Verify installation
docker --version
```

Configuration Files

1. Create docker-compose.yml:

services:

app: # Main application service

image: ghcr.io/krait0/najahi:latest # Change this to your image repository

build:

context: . # Build context is current directory

dockerfile: Dockerfile # Points to Dockerfile in root

ports:

- "3000:3000" # Map host port 3000 to container port 3000 (adjust if needed)

environment:

Email Configuration

EMAIL_USER: ""

EMAIL_PASS: ""

EMAIL_TO: ""

NEXT_PUBLIC_CONTACT_API_ROUTE: "/api/contact"

Cloudflare Turnstile

NEXT_PUBLIC_CLOUDFLARE_TURNSTILE_SITE_KEY: ""

CLOUDFLARE_TURNSTILE_SECRET_KEY: ""

Analytics

NEXT_PUBLIC_GA_ID: "your-google-analytics-id"

NEXT_PUBLIC_BING_ID: "your-bing-id"

NEXT_PUBLIC_YANDEX_ID: "your-yandex-id"

Application

NODE_ENV: "production"

PORT: "3000"

Security

CORS_ORIGINS: "https://yourdomain.com,https://api.yourdomain.com"

restart: unless-stopped # Restart policy (options: no, always, on-failure, unless-stopped)

networks:

- najahi-network # Connect to custom network (defined below)

volumes:

- najahi-uploads:/app/public/uploads

- najahi-data:/app/data
- najahi-cache:/app/.next/cache

networks:

najahi-network: # Custom network definition
driver: bridge # Network driver type (bridge is default)

volumes:

najahi-uploads:
najahi-data:
najahi-cache:

2. Create Dockerfile:

Stage 1: Dependencies

FROM node:20-alpine AS deps

LABEL org.opencontainers.image.source="https://github.com/kraitto/najahi"

LABEL org.opencontainers.image.description="Najahi - Personal Portfolio Resume
Template Developed by Kraitto"

LABEL org.opencontainers.image.licenses="Commercial License"

RUN apk add --no-cache libc6-compat

WORKDIR /app

Copy package files

COPY package.json package-lock.json* ./

COPY .env .env.local* ./

Install dependencies

RUN npm install

Stage 2: Builder

FROM node:20-alpine AS builder

WORKDIR /app

ENV NEXT_TELEMETRY_DISABLED=1

ENV NODE_ENV=production

ENV SKIP_ENV_VALIDATION=1

ENV SKIP_SMTP_VERIFY=true

COPY --from=deps /app/node_modules ./node_modules

COPY --from=deps /app/.env* ./

COPY . .

Build the application with increased memory limit

RUN node --max_old_space_size=4096 node_modules/.bin/next build

Stage 3: Runner

FROM node:20-alpine AS runner

WORKDIR /app

ENV NODE_ENV=production

ENV NEXT_TELEMETRY_DISABLED=1

RUN addgroup --system --gid 1001 nodejs

RUN adduser --system --uid 1001 nextjs

Copy necessary files

COPY --from=builder /app/public ./public

COPY --from=builder /app/.next/standalone ./

COPY --from=builder /app/.next/static ./next/static

COPY --from=builder /app/.env* ./

Create cache directory and set permissions

RUN mkdir -p .next/cache

RUN chown -R nextjs:nodejs .next

RUN chmod -R 755 .next

USER nextjs

EXPOSE 3000

ENV PORT=3000

ENV HOSTNAME="0.0.0.0"

CMD ["node", "server.js"]

Deployment Steps

1. Start the containers:
docker-compose up -d
2. Verify deployment:
docker-compose ps
docker-compose logs

Method 3: PM2 Deployment

Installation & Setup

1. Install PM2:
npm install -g pm2
2. Install dependencies:
npm install
3. Build application:
npm run build

Configuration

Create ecosystem.config.js:

```
module.exports = {  
  apps: [  
    {  
      name: "najahi",  
      script: "./node_modules/next/dist/bin/next",  
      args: "start",  
      //instances: 'max',  
      exec_mode: "cluster",  
      autorestart: true,  
      watch: false,  
      max_memory_restart: "1G",  
      env_production: {  
        NODE_ENV: "production",  
        PORT: 3000,  
      },  
      env_development: {  
        NODE_ENV: "development",  
        PORT: 3000,  
      },  
    },  
  ],  
};
```

Deployment Steps

1. Start application:
pm2 start ecosystem.config.js --env production
2. Setup auto-restart:
pm2 startup
pm2 save
3. Monitor application:
pm2 monit
pm2 logs najahi

Troubleshooting Guide

Build Failures

1. Clear build cache:
rm -rf .next
npm run build
2. Check dependency conflicts:
npm ls

Runtime Errors

1. Check logs based on deployment method:
Vercel
vercel logs

Docker
docker logs najahi

PM2
pm2 logs najahi
2. Common fixes:
 - Restart application
 - Verify environment variables
 - Check system resources
 - Monitor memory usage

Performance Issues

1. Monitor metrics:
PM2
pm2 monit

Docker
docker stats najahi
2. Optimization steps:
 - Scale instances
 - Adjust memory limits
 - Enable caching

Security Checklist

- Environment variables properly set
- CORS origins configured
- SMTP credentials protected
- SSL/TLS enabled
- Regular security updates
- Proper error handling