

KHALED GAMAL BIN NAQEEB

WEB DEVELOPER

SUMMARY

Highly motivated and skilled React.js developer with 6 months of experience in building real-world applications. Seeking a challenging position where I can utilize my expertise in React.js, Node.js, Express.js, and Nest.js to contribute to the success of a dynamic development team.

QUICK BITS . . .

- Highly motivated
- Optimistic
- Passionate
- Avid learner
- #React.js and #Node.js enthusiast

CONTACT

✉ nakqeeb@gmail.com

☎ +91 8197991938

in <https://www.linkedin.com/in/nakqeeb>

🌐 <https://github.com/nakqeeb>

EXPERIENCE

React.js Developer

Neumtech Jan 2023 - Jun 2023

- Worked on **Rakaya** project for monitoring food quality.
- Developed and implemented user-friendly and responsive web applications using React.js.
- Collaborated with cross-functional teams to define project requirements and deliver high-quality software solutions.
- Conducted code reviews and ensured adherence to coding standards and best practices.

SKILLS

- Programming Languages: JavaScript, TypeScript, HTML, CSS
- Frameworks/Libraries:
 - React.js (6 months of experience)
 - Familiar with Node.js, Express.js, Nest.js (easily recover knowledge)
- Version Control: Git, GitHub
- Database: MongoDB, PostgreSQL
- Tools: Bootstrap, Redux, Firebase, Visual Studio Code, Postman, Chrome DevTools
- Documented APIs using OpenAPI Specification (Swagger)
- Quick learner with a strong ability to adapt to new technologies.

EDUCATION

Bachelor of Computer Applications

Bangalore University Jun 2016 - Jun 2019

PROJECTS

Portfolio

Interactive Portfolio Application with ReactJS

This project showcases your skills in modern web development and design by building a personal portfolio website.

Key features:

- **ReactJS:** Utilizes React, a popular JavaScript library, for building a dynamic and interactive user interface.
- **Figma Inspiration:** Design inspired by Figma, demonstrating your ability to translate design mockups into functional code and strong design sensibilities.
- **Portfolio Functionality:** Likely includes sections showcasing your work experience, skills, projects, and potentially contact information.

This application demonstrates expertise in:

- ReactJS development
- Translating Figma designs into functional code
- Building user interfaces with a focus on user experience

<https://github.com/nakqeeb/portfolio>

Take a look at ([My Portfolio](#))

[Figma Design](#)

Candleaf (E-Commerce App)

Frontend Application with ReactJS, Redux Toolkit, and Vanilla CSS

This frontend application builds upon the previous description by showcasing additional functionalities and advanced development practices:

Key features:

- **ReactJS:** Utilized for building user interfaces with efficient component-based architecture.
- **Redux Toolkit:** Leverages Redux Toolkit, a modern wrapper around Redux, for managing application state in a more efficient and streamlined manner.
- **Pure CSS styling:** No external styling libraries are used, relying on Vanilla CSS for complete control over the visual style.
- **Figma inspiration:** Design inspiration drawn from Figma, indicating strong design sensibilities and the ability to translate mockups into functional code.

This application demonstrates expertise in:

- ReactJS development
- Redux Toolkit for state management
- Writing clean and maintainable Vanilla CSS

LANGUAGES

Arabic

English

- Design inspiration and implementation
- Building modern and performant web applications with advanced state management capabilities

<https://github.com/nakqeeb/candleaf>

Take a look at ([Candleaf app](#))

[Figma Design](#)

HTML and CSS websites (No frameworks)

Three Hand-Crafted Websites using HTML & Vanilla CSS.

Key features:

- Developed three customizable HTML and Vanilla CSS templates with a focus on clean code structure, user experience, and responsiveness for optimal performance across devices. These templates can be adapted for various website layouts (e.g., landing page, portfolio, blog).
- Implemented semantic HTML for clear content organization and accessibility.
- Demonstrated strong understanding of CSS fundamentals for effective styling.
- Utilized responsive design principles to ensure optimal layout across various screen sizes.

Source code:

[HTML Template 1](#) - [HTML Template 2](#) - [HTML Template 3](#)

Take a look:

[Demo 1](#) - [Demo 2](#) - [Demo 3](#)

Awesome Places

A Full-Stack application allows users to manage their favorite places in one convenient location.

Features:

- User Authentication: Sign-up and login functionality.
- Place Management:
 - Add new places with details and location using Google Maps integration.
 - Edit existing places.
 - Delete unwanted places.
- Interactive map: View all added places displayed on a map for easy navigation and visualization.
- Responsive design: Adapts to different screen sizes for optimal user experience.

Technology stack:

- Frontend: ReactJS.
- Backend: NodeJS & ExpressJS.
- Database: MongoDB.
- Other: Google Maps API

Frontend:

<https://github.com/nakqeeb/awesome-places-frontend>

Backend:

<https://github.com/nakqeeb/awesome-places-backend>

ToDo App

Backend Application with NestJS, MongoDB, and CRUD Operations

This backend application utilizes the NestJS framework to provide a robust and feature-rich foundation.

Features:

- CRUD Operations: Create, Read, Update, and Delete data effectively.
- Authentication: Secure user authentication and access control.
- Guards: Implement custom guards to restrict access to specific routes.
- ValidationPipe: Ensure data integrity and validation before processing requests.
- MongoDB Integration: Leverage the powerful MongoDB database for efficient data management.
- Mongoose ODM: Benefit from Mongoose's Object-Document Mapping capabilities for smooth data interaction.

Technology stack:

- NestJS framework
- MongoDB database
- Mongoose ODM library

<https://github.com/nakqeeb/todo-nestjs>

Same App (ToDo App) with different technologies:

Advanced Backend Application with NestJS, PostgreSQL, GraphQL, and Error Handling

This backend application leverages the NestJS framework, PostgreSQL database, and GraphQL API to deliver a powerful and feature-rich solution.

Features:

- Advanced CRUD Operations: Create, Read, Update, and Delete data with enhanced capabilities.
- PostgreSQL Integration: Utilize PostgreSQL's robust features and strong data security.
- Prisma Database Toolkit: Simplify data interaction and schema management with Prisma.
- GraphQL API: Design an efficient and flexible API using GraphQL schemas, resolvers, and queries.
- Apollo Server Integration: Leverage Apollo Server for smooth GraphQL API implementation.
- Error Handling and Logging: Track and capture exceptions, errors, and performance issues with Sentry's SDK.

Technology stack:

- NestJS framework
- PostgreSQL database
- Prisma database toolkit
- GraphQL API with Apollo Server
- Sentry SDK for error handling and logging

<https://github.com/nakqeeb/todo-prisma-graphql-nestjs>

Posts App

NestJS Backend Application with CRUD, Authentication, and PDF Export

This NestJS backend application provides a comprehensive set of features for data management and user authentication.

Features:

- CRUD operations: Create, Read, Update, and Delete data efficiently.
- Authentication: Secure user authentication and access control.

- Guards and roles: Implement custom guards and roles to restrict access based on user permissions.
- ValidationPipe: Ensure data integrity and validate user input.
- PDF Export: Generate and export data to PDF format for offline access or sharing.
- SQLite Database Integration: Leverage the lightweight and easy-to-use SQLite database with TypeORM.
- Object-Relational Mapping (ORM): Benefit from TypeORM's capabilities for smooth data interaction and manipulation.

Technology stack:

- NestJS framework
- SQLite database
- TypeORM ORM library

<https://github.com/nakqeeb/posts-nestjs>

Notes App

NodeJS Backend Application with ExpressJS, MongoDB, and CRUD Operations

This backend application leverages the NodeJS and ExpressJS frameworks to build a robust and flexible REST API.

Features:

- CRUD Operations: Create, Read, Update, and Delete data efficiently.
- ExpressJS Integration: Utilize ExpressJS for routing and handling API requests.
- MongoDB Integration: Leverage the powerful MongoDB database for efficient data management.
- Mongoose ODM: Benefit from Mongoose's Object-Document Mapping capabilities for smooth data interaction and manipulation.
- Authentication: Implement user authentication and access control to secure data access.

Technology stack:

- NodeJS
- ExpressJS
- MongoDB database
- Mongoose ODM library

<https://github.com/nakqeeb/note-app-backend>