

Om Singh

Bengaluru, Karnataka | omsingh2849@gmail.com | +919508732458 | My Website

linkedin.com/in/om-singh-0404bx | github.com/om-singh-D

About Me

I'm a passionate and curious Computer Science student with a strong foundation in Data Structures and Algorithms using C++. I thrive on solving problems and building real-world applications using the MERN (MongoDB, Express.js, React, Node.js) and PERN (PostgreSQL, Express.js, React, Node.js) stacks. With hands-on experience in full-stack development, I enjoy bringing ideas to life through clean code, intuitive UI, and scalable backend logic. I aspire to grow as a Software Engineer, contributing to impactful projects while continuously learning cutting-edge technologies.

Currently exploring system design, RESTful APIs, and modern web architectures. Open to internships and projects where I can learn, collaborate, and deliver meaningful software solutions..

Skills

- **Data Structures and Algorithms (C++):** Proficient in problem-solving, logic building, and algorithmic thinking.
- **Full-Stack Development (MERN):** Experience building dynamic web apps using MongoDB, Express.js, React, and Node.js.
- **Frontend and UI Design:** Skilled in creating responsive, user-friendly interfaces using React, HTML, CSS, and Tailwind.
- **Backend and API Integration:** Capable of designing RESTful APIs, handling authentication, and integrating PostgreSQL/MongoDB databases.

Education

Dayananda Sagar University, B.Tech in CSE(Data Science)

2023 – 2027

- GPA: 8.1/10
- **Coursework:** DAA, DBMS, COA, OS

Projects

MedVault

github.com/om-singh-D/e-health-record-system-MedVault

- Comprehensive healthcare management platform connecting patients and doctors. Features include secure user authentication, detailed patient medical profiles, doctor specialization management, prescription tracking, medical history records, and appointment scheduling. Built with MongoDB for robust data storage and Node.js backend.
- Tools Used: MERN Stack

ChefBot: Generative AI Recipe Assistant

Certification

- Developed and deployed a cloud-native web application that generates personalized meal recommendations and recipes using Google's Gemini generative AI model. The app allows users to specify cuisine type, dietary preferences, allergies, available ingredients, and wine preferences, and then provides detailed recipes with nutritional information and wine pairings. Built with Python and Streamlit, the application leverages Google Vertex AI for natural language generation, Google Artifact Registry for container storage, and Cloud Run for scalable deployment.
- Tools Used: Python, Streamlit, Vertex AI, Google Cloud Run, Docker

Personal Portfolio

github.com/om-singh-D/Portfolio

- A modern, responsive portfolio website using Next.js 15, TypeScript, and Tailwind CSS to showcase technical

skills and professional experience. The site features a sophisticated UI with glassmorphic design elements, custom animations powered by Framer Motion, and interactive particle effects using HTML5 Canvas. Implemented a fully functional contact form with real-time validation, toast notifications, and email service integration, along with smooth scroll navigation, mobile-responsive design, and a custom toast notification system. Built with component-based architecture using 15+ reusable React components, custom hooks for mobile detection and notifications, and optimized for SEO with proper metadata. The project demonstrates proficiency in modern web development technologies, responsive design principles, and attention to both functionality and user experience, resulting in a production-ready application with optimized build size and performance.

- Tools Used: Next.js, Typescript

Nutritional Deficiency Analyzer

- The Nutritional Deficiency Analyzer is a MATLAB GUI program built to help users analyze if their daily food intake meets the suggested intake levels for six essential nutrients: Protein, Vitamin C, Calcium, Iron, Vitamin D, and Potassium. The program takes the user input as food names and corresponding quantities in grams. When the 'Analyze' button is clicked, the inputs are processed using a pre-loaded dataset (Nutrients.mat) which cross-references each item of food to standardised nutrient values per 100g, based upon USDA FoodData Central nutrient IDs.
- Tools Used: MATLAB

Technologies

Languages: C++, C, SQL, JavaScript, TypeScript, Python

Technologies: REACT, Next.js, Tailwind CSS / Bootstrap, Redux, Node.js, Express.js, MongoDB, PostgreSQL

Tools: Git and GitHub, VertexAI, Postman, REST APIs, npm / yarn