Siddharth Behera

devsiddharthbehera@gmail.com | +91-8455932162 |devsiddharthbehera.in

PROFILE

I'm a results-driven Software Engineer with over 3 years of experience building scalable, high-performance web applications using the MERN stack. I'm passionate about crafting intuitive user experiences with React (Redux, Router) and developing reliable backend systems with Node.js and Express.
Recently, I've been expanding my expertise into System Design and modern software architectures — including Microservices, Docker/Kubernetes, and event-driven systems — to build solutions that are not only fast, but also resilient and future-ready. I thrive on solving complex problems, optimizing performance, and collaborating to turn ideas into impactful digital products.

WORK EXPERIENCE

Software Engineer | Vortex Digital Solutions Oct 2022 – Present

- Led the development of scalable full-stack features using the MERN stack (MongoDB, Express, React, Node.js), managing end-to-end delivery from database modeling to responsive frontend implementation.
- Optimized application performance by implementing compound MongoDB indexes and strategically refactoring Mongoose aggregation pipelines, resulting in a 70% reduction in average query time for critical reporting features.
- Implemented a resilient, high-availability data layer by integrating Redis caching for frequently accessed, high-read endpoints, ensuring sub-500ms response times for the main dashboard and eliminating server crashes under peak load.
- Architected and developed RESTful APIs using Node.js and Express.js, securing them with JWT authentication and bcrypt.js for password hashing, ensuring robust data security and access control.
- Collaborated on establishing CI/CD pipelines using GitHub Actions, automating build and deployment processes to staging and production environments, resulting in a 40% reduction in manual deployment errors.
- Contributed to the design and initial setup of a Microservices architecture project, gaining practical experience with Docker for containerization and foundational knowledge of Kubernetes for orchestration.

PROJECT

Chat Application – "Talk-A-Tive"
Deployment: Render
Link: https://talk-a-tive-pwvu.onrender.com/
Description:
Talk-A-Tive is a real-time chat web application built using the MERN stack with secure user authentication. It allows users to chat one-on-one or in groups, create channels, and manage participants. The application leverages WebSockets via Socket.io for seamless, real-time messaging and provides a responsive interface for both desktop and mobile users.
Summary:
Developed a full-featured real-time chat application with secure user authentication, personal and group chat management, and user search capabilities. The app supports adding or removing members, renaming group chats, and includes real-time notifications with typing indicators. Designed with a responsive and fluid user interface optimized for both desktop and mobile devices.

EDUCATION

Central Tool Room & Training Centre (CTTC), Bhubaneswar – Post-Graduate Diploma, Tool Design and CAD/CAM
Central Institute of Plastics Engineering & Technology (CIPET), Bhubaneswar – B.Tech, Plastic Engineering

TECH STACK

Frontend	HTML5, CSS3, JavaScript, TypeScript
Libraries & UI	React (Redux, React Router), Bootstrap, Chakra UI, Hope UI, Axios, Socket.io-client
Backend	Node.JS, Express.JS, JWT, Bcrypt.js, Socket.io
Database	MongoDB, Redis
ORM/ODM	Mongoose
DevOps / Microservices	Docker, Kubernetes, Kafka
CI/CD	GitHub Actions
Tools & Utilities	Git, GitHub, Postman, Firebase
IDE / Editor	Visual Studio Code

SOFTWARE ARCHITECTURE & DESIGN CONCEPTS

As an emerging professional, I have developed a good understanding of critical concepts and tools, including...

- Networking Basics: Understanding of HTTP/HTTPS protocols, REST APIs, WebSockets, DNS resolution, and TCP/UDP communication
- Software Architecture: Understanding Microservices, and Monolithic vs. Distributed architectural patterns
- Nginx: Basic configuration, reverse proxy, load balancing
- Containerization & Orchestration: Managing Docker containers and orchestrating Kubernetes clusters and pods for scalable deployments
- Microservices architecture: Understanding service decomposition, inter-service communication, and scalability benefits
- DevOps Principles: Implementation of Continuous Integration (CI), Continuous Deployment (CD), and Infrastructure as Code (IaC) practices for streamlined development and delivery
- System design Fundamentals: Designing for scalability (vertical and horizontal scaling), fault tolerance, high availability, reliability, redundancy, and optimal throughput
- Messaging systems: Understanding of Message Queues (e.g., Kafka) and event-driven architectures for asynchronous communication
- Security Fundamentals: Authentication, Authorization, JWT, Encryption
- Database Concepts: Strong understanding of SQL and NoSQL databases, ACID transactions, data modeling, indexing strategies, caching with Redis, as well as replication and sharding for scalability and reliability
- Advanced Architectural Concepts: Understanding of CAP and PACELC theorems, consistent hashing for distributed systems, proxies and load balancers for traffic management, and CDN for content delivery optimization
- API Design & Communication Paradigms: Designing and integrating APIs using REST, GraphQL, and gRPC, with a focus on backward compatibility and maintainability
- Monitoring & Observability: Implementing logging, metrics, and alerting systems, with integration into external monitoring services for performance tracking and issue resolution
- Object Storage & External Services: Designing and integrating scalable object storage systems and cloud-based external APIs for data persistence and interoperability

PROJECT
Connect-X – Real-Time Communication Platform
Deployment: Render
Link: https://connet-x674.onrender.com
Description:
Connet-X is an ongoing personal project built with a strong emphasis on system design, scalable microservices, and real-time communication. The project was rapidly prototyped using GitHub Copilot AI, allowing more focus on architecture, distributed communication patterns, and performance considerations.

Key Features:

- Google OAuth for secure authentication
- Peer-to-peer (P2P) file transfer using WebRTC
- High-quality real-time video calling
- Live user presence and messaging powered by Socket.io
- Profile picture upload and update
- Separated services to reflect microservices patterns (auth, communication, file handling, etc.)

Summary:
Designed as a learning-focused yet production-like application to deepen expertise in system design, WebRTC, and event-driven real-time communication workflows. The platform showcases scalable architecture choices, efficient state handling, and seamless real-time interactions.