

Ishwer Singh

[GitHub](#)

[LinkedIn](#)

[Email](#)

Amritsar, Punjab, India

Summary

Detail-oriented and motivated Computer Science student pursuing a B.Tech (2022–2026), with a solid foundation in programming and software development. Passionate about building efficient, user-friendly, and scalable web applications using modern technologies.

Skills

Languages: JavaScript (ES6+), TypeScript, Python, SQL, C++, CSS, Java, Shell Scripting (Bash).

FrontEnd Development: React, Redux, Next.js, HTML, CSS, Responsive Design, Ionic

BackEnd Development: Node.js, Firebase Auth

Development & Editing Tools: VS Code, Git, Postman, Figma, Canva, CapCut, Adobe Photoshop, Webpack

Concepts: Distributed Systems, System Monitoring, Incident Management, Service-Oriented Architecture, Microservices, Security Best Practices, Performance Optimization, Agile Methodologies

Tools: Postman, Figma, OpenAI

Soft Skills: Ownership, Communication, Team Collaboration, Problem Solving, Adaptability, Creativity

Education

Bachelor of Technology in Computer Science & Engineering

Amritsar College of Engineering & Technology (GPA: 5.5/10)

Amritsar, India

Senior Secondary

Khalsa college Public School (percentage - 55%)

Punjab, India

Projects

SecureShare | a Web Application

- Technology Stack: React, Redux, Python, Django, SQLite, Docker, Web Crypto API, JWT Authentication, AES-256 Encryption
- Built a secure file-sharing platform with **MFA, RBAC, and AES-256 encryption** for temporary, shareable links.
- Implemented **AES-256 encryption** for file uploads, creating secure, shareable links with expiration features to provide controlled and temporary file access
- Enforced **cybersecurity best practices** including **HTTPS with TLS, bcrypt password hashing, JWT session management, and comprehensive input validation** to safeguard against vulnerabilities.
- Delivered a clean, modular full-stack solution, ensuring rigorous encryption, seamless client-server communication, and **Dockerized deployment** for scalability and ease of use in production environments.

WellQuest | a Native Mobile App

- Technology Stack: React Native, Expo, MongoDB, AWS EC2, JWT Authentication, OpenAI, Google Vision API
- **Designed and developed responsive, user-friendly mobile interfaces** using **React Native and Expo**, significantly enhancing client engagement and retention.
- **Built and integrated progress-tracking tools** that allow gym trainers to monitor clients' performance metrics, providing personalized insights and enhancing user motivation and retention.
- **Developed back-end services using MongoDB** for efficient data storage and **AWS EC2** for scalable cloud hosting, ensuring seamless operation and performance.
- Integrated advanced APIs, such as **OpenAI** for intelligent insights and **Google Vision API** for **image recognition**, delivering a feature-rich and seamless user experience.

PawGuard | a Web Application

- Technology Stack: React, Express, MongoDB, Node.js, Microsoft IIS Server, TomTom API

- Developed reusable, custom React components and optimized the user interface for improved responsiveness and accessibility, significantly enhancing user satisfaction.
- Engineered seamless integration between the React front-end and Node.js/Express back-end, ensuring efficient data flow and improving application reliability and performance.
- Designed and implemented a comprehensive database system using MongoDB, enabling efficient management of pet insurance data and streamlined clinic selection for pet owners.
- Integrated TomTom API for interactive maps, enhancing the clinic selection process by providing real-time geolocation and routing capabilities for pet owners.

SmartPack | a Progressive Web App

- Technology Stack: HTML, CSS, JavaScript, Firebase, Service Workers, Web Push Notifications, PWA Best Practices
- Developed SmartPack, a **Progressive Web App (PWA)** for task and activity planning, improving user organization and goal tracking.
- Implemented **task management features**, including dynamic list creation, item addition, and real-time progress tracking.
- Used **Firebase** for seamless backend integration and real-time synchronization across devices.
- Ensured **offline functionality** through service workers, enabling users to access tasks without an internet connection.