CONTACT

Phone
+92-311-0846554
Email
mrameezch16@gmail.com
Address
Lahore, Punjab

EDUCATION

Bachelor of Computer Science
Superior University

EXPERTISE

- React.js
- Next.js
- TypeScript
- JavaScript
- Redux Toolkit / RTK Query
- Tailwind CSS
- ShadCN/UI
- HTML5
- CSS3
- PostgreSQL
- Git
- GitHub

AI TOOLS & CAPABILITIES

Advanced, daily use for coding, debugging, and shipping features:

- Cursor
- Windsurf
- GitHub Copilot
- Lovable
- Codex

Regular use for research, planning, and content:

- Claude
- Gemini

MCP INTEGRATIONS

Configured and used MCP servers to extend AI tools into real dev workflows:

- Figma MCP — design-to-code handoff
- GitHub MCP — repo, PR & issue management
- PostgreSQL / Database MCP — schema & query access
- Slack / Jira MCP — task & team sync

LANGUAGE

English
Urdu

M. RAMEEZ SAJJAD

SOFTWARE DEVELOPER

Detail-oriented developer with 1 year of hands-on, professional experience in front-end and back-end development. Proficient in React, Next.js, TypeScript, JavaScript, and PostgreSQL, with a strong focus on building scalable, responsive, and user-friendly web applications. Comfortable working with modern AI-assisted development tools and MCP integrations to accelerate delivery. Passionate about writing clean, maintainable code and contributing to real-world software products.

EXPERIENCE

- Developed and maintained web applications using React, TypeScript, HTML, CSS, and JavaScript.
- Built and integrated RESTful APIs using Node.js and Express.js.
- Designed and managed relational databases using PostgreSQL, including CRUD operations.
- Implemented reusable React components and ensured responsive UI across devices.
- Collaborated with team members using Git/GitHub for version control and code reviews.
- Debugged and optimized application performance and backend queries.
- Used AI-assisted tools (Cursor, Windsurf, Copilot, Codex) and MCP integrations (GitHub, PostgreSQL, Figma) to speed up development, code review, and debugging.

PROJECTS

SociiLabs

Prophic AI

Tech Stack: React.js, TypeScript, Redux Toolkit, RTK Query, Tailwind CSS, ShadCN/UI, Clerk

Developed and enhanced production-ready features for Prophic AI, an AI-powered business application, working across UI design, frontend development, feature implementation, and UX improvements.

- Designed and implemented responsive, reusable interfaces using React.js, Tailwind CSS, ShadCN/UI, and Radix UI, maintaining consistency across the application.
- Built a comprehensive notification center with filtering, pagination, notification-type handling, human-readable messages, localization, and intelligent routing to relevant application workflows.
- Implemented and improved configuration template and service-role management workflows, including side-drawer based create/update interfaces, translations, and reusable UI components.
- Optimized application search and data fetching by implementing debounced search, pagination improvements, and reduced unnecessary API polling/calls, improving frontend performance and responsiveness.
- Implemented secure password-change re-verification flows using Clerk, strengthening authentication and account security.
- Worked extensively with Redux Toolkit, RTK Query, React Router, React Hook Form, Zod, i18next, and REST APIs to build scalable and maintainable frontend functionality.
- Collaborated on bug fixes, UX refinements, localization, component refactoring, and application-wide improvements while maintaining existing architecture and reusable components.

Prophic Marketing Website

Tech Stack: Next.js, React, JavaScript, Tailwind CSS, Motion, SendGrid, REST APIs

- Developed a complete, production-ready marketing website from scratch using Next.js 15, React, and Tailwind CSS, covering the full development lifecycle from UI implementation to functionality and integration.
- Built a scalable reusable component architecture with shared UI components, responsive layouts, navigation, CTAs, FAQs, pricing cards, integration sections, forms, and page-specific components.
- Developed multiple responsive pages including Home, Features, How It Works, Pricing, Demo, About, Contact, Privacy Policy, and Terms of Service.
- Implemented polished interactive animations and transitions using Motion, including section reveals, hover interactions, animated UI elements, marquees, and CTA effects.
- Built a dynamic pricing calculator and pricing comparison interface with API-driven pricing functionality.
- Implemented a functional contact form API with SendGrid integration, input validation, consent handling, email delivery, and server-side error handling.
- Added backend security measures including IP-based rate limiting, honeypot bot protection, request validation, and secure client-IP resolution.
- Optimized the application for responsive design, SEO, accessibility, maintainability, and production deployment using the Next.js App Router.

Adeptis Marketing Website

Tech Stack: Next.js 15, React 19, Tailwind CSS v4, JavaScript

- Implemented the website architecture, UI components, page layouts, navigation, responsive behavior, typography, design tokens, and reusable component system.
- Developed a complete, responsive multi-page marketing website from scratch using Next.js 15, React 19, Tailwind CSS v4, and JavaScript.
- Built and integrated multiple sections and pages including Home, Services, Solutions, About, Case Studies, Blog, FAQs, Contact, Builder, and Quick Check.
- Developed reusable UI components for buttons, navigation, cards, typography, FAQs, CTAs, forms, comparison sections, and other recurring design patterns to improve consistency and maintainability.
- Implemented functional contact and booking forms with server-side API handling, input validation, email integration through SendGrid, rate limiting, and spam protection.
- Established a structured design system with custom typography, spacing, colors, borders, responsive layouts, and Tailwind CSS theme tokens.
- Optimized the application using Next.js App Router, reusable components, responsive design practices, and production-ready project structure.

Vimzy - WordPress Maintenance & Services Platform Marketing

Tech Stack: Next.js, React, JavaScript, Tailwind CSS, Motion

- Built a production-ready WordPress maintenance and services platform from scratch, independently handling the architecture, UI implementation, responsive layouts, reusable components, and feature development.
- Built the application using Next.js, React, JavaScript, Tailwind CSS, and Motion, implementing a scalable component-based architecture across landing, services, pricing, blog, affiliate, contact, emergency support, and plugin sections.
- Developed interactive features including service-order workflows, contact and emergency forms, pricing plan components, plugin documentation/search, newsletter subscriptions, booking integrations, and support flows.
- Implemented a dedicated WordPress Plugin Hub with plugin landing pages, documentation, changelogs, downloads, support forms, feedback, cross-selling, and WordPress.org statistics integration.
- Implemented technical SEO infrastructure including dynamic metadata, canonical URLs, Open Graph data, XML sitemap support, breadcrumbs, JSON-LD structured data, and SEO-focused page architecture.
- Integrated third-party services including Google Analytics, Microsoft Clarity, Crisp Chat, MailerLite, Rewardful, reCAPTCHA, and Cal.com while maintaining reusable integration components.
- Built server-side API routes for contact submissions, service orders, emergency support, newsletter subscriptions, development briefs, OG image generation, and content revalidation.
- Optimized the application for responsive design, accessibility, performance, maintainability, and reusable UI patterns across a large multi-page application.

ByteMage

AI Concierge — Management Dashboard & Embeddable Chat Widget

Role: Frontend Developer / Application Contributor | Tech Stack: React, TypeScript, Vite, TanStack Router, React Query, Axios, Tailwind CSS, Framer Motion, Recharts, Supabase, REST APIs, Cloudflare Workers

Designed and developed a production-style AI Concierge platform from scratch, combining a management dashboard for workspace configuration, knowledge-base administration, and analytics with a lightweight embeddable AI chat widget for external websites.

- Independently owned the application architecture, UI, state management, API integration, and user experience across the dashboard and widget.
- Built a complete knowledge-base management system supporting PDF/DOCX/TXT uploads, URL crawling, Q&A creation/deletion, upload progress tracking, validation, and error handling.
- Implemented Supabase authentication and onboarding flows including sign-up/sign-in, session management, profile completion, and protected application workflows.
- Developed Production/Testing environment management with API key handling, environment status, creation forms, and API-driven CRUD operations.
- Created analytics and conversation interfaces with KPI dashboards, Recharts visualizations, recent conversations, conversation transcripts, and CSAT-related insights.
- Built a configurable AI chat widget with message history, loading/error states, dynamic branding, welcome messages, widget configuration, and generated embed configuration.
- Exposed a lightweight window.ChatWidget.init() API and optimized the widget as an IIFE bundle for CDN distribution and simple <script>-tag integration.
- Implemented persistent conversation sessions using browser session storage and backend session retrieval, with session-aware knowledge-base queries, current-page context, structured response validation, and source citations with relevance scores.
- Developed a dynamic citation/source interface and remote widget configuration for workspace-specific brand colors, greetings, voice/CSAT settings, and other runtime options without rebuilding the widget.
- Added an optional browser-side page-content capture pipeline that extracts rendered DOM content, detects SPA route/view changes, removes irrelevant markup, deduplicates captures, and sends content to a knowledge-base ingestion endpoint.
- Added a dedicated Cloudflare Workers deployment layer for serving the generated widget bundle.
- Implemented responsive and accessible UI components using Tailwind CSS, Radix UI, form validation, dark/light themes, animations, loading states, and toast-based feedback.

Multi-Tenant Feature Flag Admin Panel

Role: Frontend & Backend | Tech Stack: React (Vite), TypeScript, Tailwind CSS, TanStack React Query, REST APIs, Chart Libraries | Domain: SaaS / Security Company Admin Platform

Developed a scalable admin panel for managing feature flags across multiple tenants for a security-focused SaaS platform. The system enables centralized control of tenant features, detailed audit trails, and real-time analytics, ensuring transparency, security, and operational efficiency.

- Integrated secure authentication and login APIs with backend services.
- Built the project from scratch using Vite + React + TypeScript, ensuring high performance and maintainability.
- Implemented Tailwind CSS for responsive and modern UI design.
- Used TanStack React Query for efficient API data fetching, caching, and state synchronization.
- Developed tenant-based feature flag management, allowing enable/disable of features per tenant.
- Integrated tenant audit trail APIs, capturing:
  - Feature on/off changes
  - User actions
  - Timestamped logs for compliance and tracking
- Added audit trail visibility within tenant and feature management forms.
- Integrated multiple statistics and analytics APIs, including:
  - Weekly/monthly scheduled hours per tenant
  - Mobile vs web user counts
  - Currently logged-in users per tenant
  - Total sites per tenant
  - Trend-based analytics
- Implemented interactive dashboards and charts to visualize tenant statistics.
- Contributed to backend-related workflows:
  - Tenant active/inactive status handling

    - Allowed tenant features configuration
    - CRUD operations for tenant features
    - Feature-flag configuration loading per tenant
    - Automatic creation of company and super admin on tenant onboarding
- Ensured scalability and modular architecture suitable for enterprise environments.

Responsive Frontend Landing Page (React + TypeScript)

Tech Stack: React, TypeScript, Tailwind CSS, HTML5, CSS3, Animations

- Built the entire landing page using React with TypeScript for type safety and scalability.
- Implemented Tailwind CSS for utility-first, responsive styling.
- Added smooth animations and transitions to enhance user engagement.
- Ensured cross-browser compatibility and mobile-first responsiveness.
- Optimized layout, typography, and spacing for a professional, production-ready look.
- Structured components for reusability and maintainability.
- Focused on performance optimization and clean semantic HTML.