

Jamshaid Mehmood

Frontend Engineer | React · Next.js · Redux | Node.js · Django REST

Lahore, Pakistan · +92 304 5297828 · jamshaidmehmood1234@gmail.com · [Portfolio](#) · [GitHub](#) · [LinkedIn](#)

SUMMARY

Frontend engineer with 2+ years on the UI layer of HireStream, a React/Next.js hiring SaaS used by enterprise recruiting teams, now leading its application-wide UI revamp. Shipped 3 role-based analytics dashboards, rebuilt 9 paginated report UIs, and led a 30+ module migration to hooks that cut technical debt by 40%. Also owns the shared component library and i18next EN/DE.

Works down the stack too: extends the Django REST Framework endpoints behind those screens — serializers, viewsets, filter backends and pagination — and ships full-stack side projects end to end, three of them deployed and publicly running.

EXPERIENCE

Frontend Developer

Jul 2024 – Present (2+ yrs)

Arbisoft, Lahore, Pakistan | HireStream — multi-tenant HR & hiring SaaS (React 17/Next.js, Redux, Django REST)

Application-Wide UI Revamp (Ongoing)

- Currently leading a revamp of the platform's entire user interface — modernising the visual language, layout and interaction patterns across every module, from Figma redesigns through to production.
- Rolling the new patterns out through the shared component library so the redesign lands consistently across dashboards, reports and workflow screens rather than page by page.

Component Architecture & Reusability

- Built a shared React component library (MetricsCard, DateRangePicker, OffsetPagination, EllipsisWithTooltip, NoteModal, RecruiterCard, FilterBar) adopted across 9 report UIs and 3 dashboards, cutting duplication and build time.

Dashboards & Reports (Figma → Production)

- Turned Figma wireframes into 3 responsive role-based dashboards (Recruiter, Team Lead, Leadership) with HiringFunnel, GenderDistribution, OfferAcceptanceTrend, RecruiterPerformanceTable and UpcomingInterviewsCard — all URL-synced.
- Rebuilt 9 report UIs (Hiring, Current Openings, Candidate Master, Offer Data, Source Performance, Time-Based Metrics, Scheduling, Referral, Custom List) with debounced search, server-side sort/filter and CSV export, URL-backed.
- Built an async export progress UI — a WebSocket-driven progress snackbar with a linear progress bar and disabled states — giving users real-time feedback during large report exports.

Backend — Django REST Framework

- Extended the DRF endpoints behind the report and dashboard surfaces — serializers, viewsets, filter backends, and pagination — so the API returned the exact shape the UI needed instead of the client reshaping payloads.
- Moved sorting and filtering server-side so large result sets stay off the client and pagination is authoritative; wrote SQL and Python scripts against the product database to verify dashboard figures and answer ad-hoc requests.

State Management & URL Sync

- Implemented Redux action/reducer/selector patterns with URL-backed filter state (search, page, page size, date range), enabling deep-link restore, per-table page-size memory, and zero-loss browser navigation across all paginated views.

Class-to-Functional Migration

- Led a platform-wide migration of 30+ modules (admin, job listing/detail, sourcing, requisitions, candidate profile, interviews, onboarding, assessments, bulk actions) from class components to hooks, cutting technical debt by 40%.

Responsive & Cross-Browser UI

- Built responsive grid layouts and mobile/desktop dual-position Job Actions menus, enforced cross-browser compatibility across Chrome, Firefox and Safari, plus a consistent chip/badge and tooltip-on-truncation pattern.
- Built the Careers & Job Discovery UI, including faceted search, bookmarks, recency labels, and similarity scores on job cards, plus step-by-step virtual tour onboarding for Settings, Jobs, Candidates, and Requisitions.

Internationalization (i18n)

- Rolled out i18next across the full frontend (EN/DE) — all new components, dynamic workflow/stage labels, and locale-aware dropdowns via a getTranslatedLabel utility — enabling the platform to serve non-English enterprise clients.

Performance & UX Polish

- Introduced skeleton loaders, debounced search inputs, and lazy-loaded chart wrappers (CustomBar, CustomDoughnut, CustomHorizontalBar via HOCs), cutting perceived load time on data-heavy report pages.
- Diagnosed and fixed a Formik re-init bug that preserved stale form state, search-highlight corruption from overlapping ranges, a filter-reset-on-refresh bug, and a double-API-call bug on rebase — reducing reported UI/UX issues by 30%.

Code Quality & Delivery

- Participated in PR reviews, enforced ESLint standards with filename context in webpack output, and shipped 20 distinct features across Agile/Scrum sprints — consistently delivering end-to-end UI from design handoff to production.

PROJECTS

SalesFlow CRM — Full-Stack Sales CRM

2026

React 19 · Express 5 · PostgreSQL · Prisma | [Live Demo](#) · [Source](#)

- Sole engineer: designed the PostgreSQL schema with Prisma, built the Express 5 REST API, and shipped the React client on top — deployed and publicly running.
- Authentication written from scratch: JWT sessions, bcrypt hashing, Google OAuth token verification, and TOTP two-factor with QR enrolment and recovery codes.
- Role-based access control enforced in middleware (admins see the organisation, reps only their own records), plus an audit log covering create/update/delete/merge across every entity.
- Analytics dashboard of KPI cards, revenue charts, deal-stage distribution and sales forecast (Recharts); drag-and-drop Kanban pipeline via DnD Kit; contacts at scale with filters, CSV import/export and duplicate merge.
- React Query for server state and caching, i18next EN/DE, persisted dark/light theme, responsive from phone to desktop.

Virtual Art Gallery — Full-Stack MERN Gallery + 3D Tour

2026

React · Tailwind · Node/Express · MongoDB Atlas · three.js · Cloudinary | [Live Demo](#) · [Source](#)

- Full-stack MERN art gallery — browse, search and filter artwork, a cart, uploads, and a walkable 3D tour.
- Walkable 3D gallery tour built on three.js, code-split into its own bundle chunk and loading artwork as WebGL textures (crossOrigin handled) so the viewer never blocks first paint.
- Cloudinary-backed upload pipeline with server-side watermarking, perceptual-hash duplicate detection, and authenticated (non-public) originals that survive redeloys.
- JWT authentication with an email-OTP signup and password-reset flow, over a Node/Express API and MongoDB Atlas.
- Deployed end to end (frontend on Vercel, API on Render) with production hardening: multi-origin CORS including a Vercel preview-URL wildcard, and fail-fast boot checks that refuse to start on missing config.

Instagram Clone — Full-Stack Photo Sharing App

2026

React 18 · TypeScript · MUI · RTK Query · Node/Express · MongoDB (GridFS) | [Live Demo](#) · [Source](#)

- Instagram-style social app — photo feed, likes, comments, follows, saved posts, search and an activity inbox.
- Keyset pagination on (createdAt, _id) rather than skip, likes made idempotent by a unique index instead of read-then-write, and photos held in GridFS so a feed page stays a few hundred bytes per post.
- Auth with 15-minute in-memory access tokens and rotating httpOnly refresh cookies that revoke the whole token family on reuse, plus Google Sign-In verified server-side; the client funnels concurrent 401s through a single refresh.
- Optimistic likes, saves and follows patched across every RTK Query cache entry with rollback, and 46 Playwright end-to-end tests covering auth, feed, profile, follow, search and responsive layout.

Fiverr Lite — Full-Stack Freelance Marketplace

2026

Next.js 14 · TypeScript · Express · MongoDB · Zod · Stripe · Socket.IO | [Live Demo](#) · [Source](#)

- Sole architect of a full v2 re-architecture of a legacy freelance-marketplace prototype: audited v1 (297 issues documented), then rebuilt a two-sided marketplace — tiered gig packages, an escrow-style order lifecycle, real-time chat, deliveries and ratings, an admin dashboard and an AI gig-copy assistant — end to end.
- Closed a systemic BOLA/IDOR class by sourcing every actor id from the authenticated JWT and deriving every order price server-side; settlement is idempotent and verified via a signed Stripe webhook plus a server-side PaymentIntent re-read, so orders can't double-charge or be marked paid on the client's word.
- Unified the stack behind a shared Zod contract (@fiverr-lite/shared) that generates both runtime validation and static DTO types for the API and web app, eliminating client/server drift.
- Enforced an escrow-style order state machine with DB-level invariants in a layered Express/TypeScript backend, and a Stripe/mock payment-provider abstraction that runs the full checkout-to-settlement flow offline; tested with Vitest + mongodb-memory-server and Playwright.

TECHNICAL SKILLS

Core Frontend: React, Next.js, JavaScript (ES6+), TypeScript, HTML5, CSS3, SASS/SCSS, Tailwind CSS, jQuery

State & Routing: Redux, React Hooks, React Query, Context API, URL-synced state, deep-link filter patterns

UI & Styling: Bootstrap, CSS Grid, Flexbox, responsive layouts, cross-browser compatibility

Data Visualization: Chart.js, Recharts, hand-written SVG charting, dashboard & KPI design

Backend: Python, Django REST Framework, Node.js, Express 5, REST API design, JWT, OAuth (Google), TOTP 2FA, bcrypt, role-based access control

Databases: PostgreSQL, Prisma, MongoDB, Mongoose, SQL (reporting & analytical queries)
Tooling: Webpack, Vite, ESLint, Git, Figma, i18next, Formik, Yup, DnD Kit, Docker (development)
Practices: Agile/Scrum, SDLC, code reviews, WCAG basics, performance optimization

EDUCATION

BS Information Technology <i>Punjab University College of Information Technology (PUCIT), Lahore · CGPA 3.7/4.0</i>	Jun 2020 – Jun 2024
Intermediate (ICS) <i>Punjab College of Information Technology (PGC), Lahore · 90%, A+</i>	Mar 2018 – Mar 2020

CERTIFICATIONS

Responsive Web Design — freeCodeCamp, 2024