Xhesi 'Jesse' Cekrezi

SENIOR SOFTWARE ENGINEER

London, England, United Kingdom
xhesi@outlook.com
07751 198697
xhesi.co.uk

KEY SKILLS

Laravel · PHP · JavaScript · TypeScript · MySQL · Elasticsearch · WordPress · C# / .NET · SignalR · Blazor · Docker · xUnit · Chrome Extensions · API Integration · Payment Integration · Internationalisation (i18n) · Automated Testing · PHPUnit · Playwright · CI/CD · Sentry · AI Coding Agents · AI Skills & MCP Connectors · RAG & LLM API Integration · Agile · Team Leadership · Problem Solving · Collaboration · Communication · User Empathy · Writing · Client Management · Azure

TECHNICAL STACK

BACKEND	Laravel, PHP 7.4 to 8.5, MySQL, SQLite, C#, ASP.NET Core, ASP.NET MVC, SignalR, Node.js, Python (Django), WordPress (custom themes and plugins)
FRONTEND	JavaScript, TypeScript, jQuery, Vue.js, Livewire, Blazor WebAssembly, Chart.js, Vite
TESTING	PHPUnit, xUnit, Playwright, Laravel Dusk, k6
TOOLS	CI/CD pipelines (GitHub Actions), Jira, Chrome Extensions (Manifest V3), Docker, Render, Azure App Service, Azure Functions, Azure Monitor, Sentry, AWS (image storage), Python scripting, AutoHotkey scripting, AI coding agents in the IDE to plan, design, implement, test and deploy, custom AI skills and Markdown instruction files and agent memory, MCP connectors linking hosting, DNS and docs systems so the whole workflow runs from one IDE, routing simpler or less urgent jobs to smaller, cheaper models running as background sub-agents alongside a larger lead agent

PROFESSIONAL SUMMARY

Senior Software Engineer with over 10 years of commercial experience building and maintaining complex web applications. Specialises in council portals and waste management systems with a focus on scalability, accessibility, and efficiency. Proven track record of leading engineering teams, improving public service platforms, managing client relationships, and delivering measurable results.

I am quick at picking up new technologies and pride myself on being a strong problem-solver. I naturally think outside the box, asking a wide range of thoughtful and effective questions to explore different angles and uncover potential solutions. This approach allows me to gather diverse ideas, test them, and communicate solutions clearly.

I stay up to date with the latest technologies and know how to use them safely and responsibly to drive innovation and growth.

EXPERIENCE

Portal Engineer Lead

AMCS, London

JAN 2020 TO PRESENT

- Lead engineer for municipality waste management portals serving millions of users, and for commercial systems including TV dashboards
- Designed and implemented scalable architecture using Laravel and modern JavaScript frameworks
- On joining in 2020, took over the existing legacy portal code, worked out how it behaved and brought it onto Laravel, doing this for four years before AI coding tools became part of my workflow
- Spearheaded accessibility improvements and modernisation efforts
- Upgraded portal codebases to newer Laravel and PHP versions
- Designed the portals to hold no client data: every record is read from and written to the back-office platform through its APIs, so nothing personal is retained on the portal side
- Generated new ideas and initiatives to sell to clients, including moving projects toward a monthly subscription model

- Took charge of departmental financials from concept through implementation and deployment
- Collaborated with cross-functional teams to deliver new features and resolve legacy system bottlenecks
- Worked with the back-office platform's API product manager, raising and agreeing the API changes the council portals I built needed
- Built payment integrations with Windcave, Worldpay, Barclaycard, Pay360, Adelante, Capita, Stripe and PayPal
- Built the portals' pricing models, with services priced by item or by time
- Built multi-language (i18n) portals: a bilingual English and Welsh portal for a Welsh council, and French, German and Spanish language demos
- Mentored junior developers, improving team efficiency and code quality
- Led a developer directly for two years and took part in hiring for the team
- Set up Azure App Service for every portal and carried out their server upgrades
- Delivered every portal through CI/CD pipelines, so sites go live safely and quickly with no downtime, and built the GitHub Actions pipeline that builds and deploys one portal to Azure App Service
- The person the team turns to when a live system goes down, though DevOps is not my role
- Used Sentry for error tracking on live portals, and built in-house error handling into the admin panels of two client portals: log viewers, system health checks, dedicated log channels that tag every entry with the user and account, and audit trails of every admin change
- Built large automated test suites for client portals, including one with over 1,100 PHPUnit tests and an 88-test Playwright suite run against the test environment
- Tested portals at every level: faked external APIs, route-by-role access matrices, Dusk and Playwright browser tests, and k6 load-test journeys
- Built and maintain the team's Chrome extension, published on the Chrome Web Store, to make the team more efficient: nobody has to ask which server a site is on, and test forms fill in seconds. It has a dashboard of 19 portals with test and live links, environment overlays, test-form autofill and bulk ticket opening, migrated to Manifest V3 and refactored around a single portal config
- Gave every portal a super admin debug mode that shows debug information on the page itself, so internal testers can see what is happening behind the scenes while APIs and other features are being tested
- Took over two legacy ASP.NET MVC (C#) council waste sites after their original developers had moved on, and now maintain them: tracing live bugs through an unfamiliar codebase, removing retired features and shipping versioned releases
- Took a client enquiry all the way to a live service on my own: proposed a standalone API in place of a core product change, ran the scoping call, wrote the specification, estimated and quoted it, then built and deployed it
- That service is a serverless Node.js API on Azure Functions that serves collection calendars to a council's own website: SOAP integration with the back-office platform, strict input validation, key authentication, rate limiting per caller and per property, client-facing documentation and a 50-case test matrix

`Laravel • JavaScript • jQuery • PHPUnit • Playwright • k6 • CI/CD • C# • ASP.NET MVC • Node.js • Payment Gateways • Chrome Extension • Azure Web Services`

Web Application Developer

Ezekia, Godalming

`JAN 2017 TO JAN 2020`

- Joined Ezekia as a startup and maintained and enhanced its executive recruitment platform, built in Laravel
- Developed a Chrome extension to integrate LinkedIn with Ezekia, streamlining recruiter workflows
- Introduced new features that improved platform adoption and user satisfaction
- Designed the MySQL database structure for new platform features, and optimised slow queries
- Used Elasticsearch for the platform's search
- Integrated the platform with Mailchimp so email campaigns could be sent from Ezekia, and stored images on AWS
- Handled incoming Stripe webhooks, verifying each signature and handling Stripe's retries

`Laravel • Vue.js • MySQL • Elasticsearch • Stripe • Chrome Extension • Mailchimp • AWS • Jira`

EDUCATION

Kingston University

BSc Computer Science, First Class Honours

2013 TO 2016 | GRADUATED WITH 81% OVERALL

Focus areas: Software Engineering, Databases, UX, Project Management

PERSONAL

I'm passionate about arts and crafts, particularly laser engraving projects that combine precision with creativity. I embrace a DIY attitude towards anything broken, whether it's hardware, software, or systems, believing that with the right approach and resources, nearly everything can be fixed or improved. With YouTube tutorials and modern language models at our fingertips, there's no skill or problem that's truly beyond reach. When I'm not engineering solutions, you'll find me bouldering, diving, exploring sports, or working on various hobbies. I hold an advanced diving certificate, and I'm a former competitive powerlifter with years of competitive experience.

Laser Engraving | DIY Projects | Bouldering | Advanced Diver | Former Powerlifter | Sports

NOTABLE PERSONAL PROJECTS

Ask about me

AI assistant on xhesi.co.uk, built with RAG and an LLM API

2026

- Built a retrieval-augmented generation (RAG) pipeline that answers visitors' questions about my experience and projects, integrated with a large language model through its official PHP SDK
- Wrote the retrieval step in PHP: the site's own CV and project content split into chunks per job and project, ranked with BM25 and synonym expansion, so every answer is grounded in real content and lists its sources
- Stored every conversation with downloadable transcripts for visitors, and reused saved answers for repeat questions so they cost no API credit
- Protected the API budget in layers: honeypot, minimum send time, a proof-of-work challenge, reCAPTCHA, per-visitor rate limits and a daily spending ceiling
- Added an admin view of chats, tokens used and estimated cost, with a test suite that fakes the AI model and a GitHub Actions check that signs in to the AI provider with Workload Identity Federation instead of an API key

Laravel 13 • PHP 8.3 • LLM API • RAG • BM25 • SQLite • GitHub Actions

Formality

Clan Home Page at formality.cc

2026 TO PRESENT

- Built and run the live home page for a 250+ member Old School RuneScape clan: roster, profiles, leaderboards, events and announcements
- Integrated Discord sign-in, forum single sign-on, and stats synced from Wise Old Man and TempleOSRS
- Built a versioned, read-only RESTful API for the clan's Discord bot to draw on: public endpoints for members, events and Boss of the Month, and key-gated endpoints for admin data, with per-key rate limits, pagination and filters
- Designed 18 scheduled jobs, staggering the heaviest refresh across eight sections to stay under API rate limits
- Delivered an admin panel with roster management, ticket triage, a health dashboard and a browser-based test runner
- Covered the application with 787 automated tests, mocking every external service

Laravel 13 • PHP 8.3 • MongoDB • SQLite • Discord OAuth • Tailwind CSS 4

Blackjack · Pick up 5 — Real-time Multiplayer Card Game for Phones

2026

blackjackpickup5.xhesi.co.uk

- Built the British street card game as a live multiplayer web game: two to six players join from their phones with a five-letter code and no sign-up
- Wrote one C# rules engine shared by the ASP.NET Core server, which enforces every move, and the Blazor WebAssembly client, which highlights playable cards
- Ran games through a SignalR hub with in-memory rooms, sending each player only their own hand so nothing can be read from the browser
- Designed for phones: portrait layout, thumb-reach controls, automatic reconnection after the screen locks and a vibration on your turn
- Covered every rule with 54 xUnit tests, including integration tests that play games through the real hub
- Containerised the server with Docker on Render and served the client as static files from shared hosting

`C# • .NET 10 • ASP.NET Core • SignalR • Blazor WebAssembly • xUnit • Docker`

The Slayer Guide — Old School RuneScape Slayer Planner

2026

slayerguide.xhesi.co.uk

- Turned a popular community Slayer spreadsheet into a web app that plans tasks for all eight Slayer masters, with live XP, GP and points rates
- Ported the workbook's formulas to PHP, including a small expression parser that evaluates the 357 formulas exported with the data
- Wrote a Python exporter that extracts 248 monster variants, 481 tasks and 232 prices from the workbook
- Proved the port with parity tests that compare the app's results against the spreadsheet's own across every monster and master
- Built saved setups with share links, JSON export and import, for anonymous visitors and accounts, with tests that keep each visitor's setups private
- Scheduled a daily refresh of loot values and item prices from the OSRS Wiki

`Laravel 12 • Livewire • PHP • SQLite • Python`

Wildy Slayer Costs — Wilderness Slayer Loot & Supply Price Tracker

2026

wildyslayercosts.xhesi.co.uk

- Built a Python and Django app that tracks what Wilderness Slayer tasks are worth, weighing loot prices against supply costs over time
- Wrote management commands that scrape 37 tasks, 86 monsters and over 4,000 drop table rows from the OSRS Wiki
- Stored hourly prices for over 4,700 Grand Exchange items, with history at hourly, six-hourly and daily resolutions
- Exported the data as JSON for a static Vue and Chart.js front end, so the public site runs no Python
- Deployed to shared hosting with a single SSH script and an hourly cron job that refreshes prices

`Python • Django • SQLite • Vue 3 • Chart.js • REST APIs`

Clan Management System — Runescape Community Platform

2016 TO 2019

- Designed and built a custom clan management system integrating Runescape, Discord, and forum member lists
- Automated membership management across platforms, ensuring consistency
- Integrated Discord to track activity, including voice chat usage and engagement metrics
- Wrote a Java script that ran in a logged-in member's game client during trips, capturing clan chat joins, leaves and damage per member and sending them to the web application, where an event log parser turned them into live attendance leaders could check mid trip
- Produced a report for every trip: headcount, peak numbers, each member's time in the chat as an attendance percentage, and combat XP gained from hiscore snapshots taken during it
- Pulled applications from the forum's REST API into a vetting queue that scored applicants on days waiting, forum posts and trips attended, and flagged inactive members by days since their last free-to-play and members trip
- Automated repetitive in-game admin with AutoHotkey: PHP pages built ready-to-run AHK scripts straight from the member database, so adding the whole roster to the friends list took one hotkey instead of hundreds of clicks
- Ran scheduled cron jobs that synced forum members through its REST API, recorded each member's daily XP from the game's hiscores and flagged name changes, merging them so a member's history followed them to the new name
- Built dashboards to identify top performers, track attendance and vet applicants, and cross-referenced who was present whenever our in-game location leaked to find spies from rival clans
- Analysed 2 years of event data for performance insights and operational security

Java • PHP • MySQL • AutoHotkey • REST APIs • Cron • Analytics

Powerlifting Management System — Final Year Project – Kingston University

2015 TO 2016

- Built a competition management platform for the Great British Powerlifting Federation in pure PHP and SQL with no framework, on a MySQL schema I designed from scratch, with separate views for the public, member athletes and the events manager
- Aggregated inconsistent regional GBPF data (PDF, Excel, Word formats) into a unified, searchable MySQL database
- Ran each competition end to end: create it by region, gym, gender, age class and capacity, invite eligible members in that region to RSVP, record weigh-ins that set each lifter's weight class, then enter all nine attempts live while the results page updated
- Automated the scoring: best attempt per lift, totals that fall to zero on a bombed lift, Wilks scores for men and women, and placings within each weight class with the lighter lifter winning a tie
- Gave athletes profiles with recent stats, records and competition history, and a rankings table filterable by region, gender, age class and weight class
- Wrote a scheduled job that moved athletes between Sub-Junior, Junior and Senior age classes and lapsed expired memberships, with test cases covering the Wilks, age class and membership rules

PHP • MySQL • PDO • jQuery • Bootstrap • Database Design

Awarded 80%+ grade (First Class standard)