

Igor Mazur

Software Developer - React / TypeScript, C# / .NET, Python, Microsoft 365

igormazur.dev

igor.j.mazur@gmail.com | github.com/ijmazur | linkedin.com/in/igor-j-mazur

Warsaw, Poland - office, hybrid or remote | Notice: three months, potentially shorter

English C2 - certified October 2026 by Eklektika (Warsaw); schooled in Ireland through the Leaving Certificate | Polish native

Five years building, shipping and supporting production software for a manufacturer of 7,000+ employees: React and TypeScript front ends, C# / ASP.NET Core on Azure, Python desktop applications, and the Microsoft 365 platform underneath much of it, which I also co-administer. I took over a system with no handover, migrated content between platforms without losing any of it, and traced a production sign-in failure to its root cause in the directory.

EXPERIENCE

Maspex Food Sp. z o.o. - Software Developer

July 2021 - present · Warsaw

A large FMCG manufacturer. Internal product engineering across web, desktop, cloud and the Microsoft 365 platform.

Full-stack web applications

- **Trade promotion platform - inherited from an external team with no handover.** Plans retail promotions, budgets the spend and books it to accounting, across about ten backend services. I ship front-end features on its React 18 / TypeScript 5 UI, and reverse-engineered the system, with agentic tooling under a citation-checked process, into the reference document the product manager and business owners now work from.
- **Sensory evaluation platform - principal contributor over two years.** ASP.NET Core and EF Core with a React / TypeScript front end on Azure, Entra ID / MSAL sign-in, running standalone and inside Microsoft Teams. Product-tasting panels scored on kiosk tablets against published statistical thresholds. When the kiosks made drag-and-drop unreliable, I replaced drag-to-rank with explicit move controls, which also made it keyboard-operable.
- **Employee onboarding platform - feature contributor.** React inside Teams over ASP.NET Core: manager-side demand editing, parent and child processes, validation and translations. For a small sister company I built the same workflow as nine Power Automate flows and a SharePoint list instead, because a full platform rollout was the wrong size for the problem.

Desktop applications - Python / Qt

- **Lead maintainer of the application that runs a factory's quality process** - incoming goods, laboratory analyses, complaints and non-conformance. PySide6 over Oracle, shipped as a Windows installer with its own self-update channel.
- **Diagnosed a production sign-in failure** that hit only users from a trusted Active Directory domain, traced it to how the directory represents those accounts, and fixed the cause with a controller-failover path rather than a workaround.
- **Five departmental applications on one shared framework** - one package layout, directory sign-on, self-updater and installer, forked per department. Sole author of the attendance kiosk, feature contributor on the grain elevator, releases and fixes on the rest; I built the skeleton one started from and mentored the junior developer on it.

Migrations

- **Legal knowledge base moved to SharePoint, sole author.** 1,029 of 1,029 articles migrated from a retiring helpdesk system by a Python pipeline against the Microsoft Graph: thirteen idempotent, re-runnable stages with a per-record log. When the editor's sanitiser kept deleting each page's author block on save, I moved that data out of the page body into library columns it cannot touch.
- **697-page Confluence wiki migration, sole author of the redo.** Rewrote a 2021 tool broken by Microsoft platform changes, rebuilt its navigation extension, re-ran it across four wikis.

Microsoft 365 platform - one of three tenant administrators for 7,000+ employees

- Governance, permissions, storage and the app catalogue, where self-service and automation are the design rule.
- **SharePoint Framework / React components I build, release and deploy:** principal developer of a newsletter builder that lets PR and HR staff send branded mailings without a designer, and of the navigation extension that serves the intranet from one data-driven component; two more adapted or extended.
- **Power Automate back ends across fourteen business areas** and three Power Apps for the legal knowledge base. About a third are called by an app or over HTTP, so the flow is an application's server side; in the legal set they call the SharePoint REST API directly where the standard connector cannot set the fields that matter.
- Planning, running and teaching the company's move off mapped network drives.

HOW I WORK

- **Write in the conventions of the codebase I am in.** React 16 class components in one product, React 18 and TypeScript 5 in another, SharePoint Framework on the toolchain the platform pins. An upgrade is deliberate work, not a side effect of a feature.
- **AI as part of the toolchain.** Within the company policies, I use agentic coding tools (Claude Code) daily across design, implementation and documentation, and configure them with project-specific skills, tool servers and scripted pipelines. Their output is a draft until it is checked against the source or the live system.

SKILLS

Languages: TypeScript, JavaScript, C#, Python, SQL (Oracle PL/SQL, PostgreSQL, T-SQL), PowerShell

Front end: React, HTML / CSS, Fluent UI, Material UI

Back end: ASP.NET Core, Entity Framework Core, REST APIs, Django REST Framework

Microsoft 365: SharePoint Online administration, SharePoint Framework (SPFx), Microsoft Graph, Power Apps, Power Automate, Teams apps, PnP PowerShell

Desktop: PySide6 / Qt, Active Directory (LDAP) sign-on

Cloud & delivery: Azure (App Service, Azure SQL, WebJobs, Entra ID / MSAL), Azure DevOps (boards, pull requests, CI/CD pipelines), Docker, Git

Data: Oracle, PostgreSQL, SQL Server / Azure SQL

Tools: Postman, Figma, Jira, Claude Code

EDUCATION & PUBLIC CODE

BSc Computer Science - Maria Curie-Skłodowska University

2019 - 2022 · Lublin

DNAstrands (github.com/ijmazur/dnastrands) - thesis project, sole-authored. A web application that encodes text into DNA sequence designs for physical data storage and generates DNA tags for marking objects, with an authenticated history and JSON export. React and Material UI over a Django REST API with JWT, PostgreSQL, Docker. Most of my professional work is closed-source; this repository, with the full thesis, is the one that can be opened and read.

I agree to the processing of personal data provided in this document for realising the recruitment process pursuant to the Personal Data Protection Act of 10 May 2018 (Journal of Laws 2018, item 1000) and in agreement with Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).