CV- ROLF VAN DAM

CONTACT

MAIL: **rolf@rolfvandam.nl (mailto:rolf@rolfvandam.nl)**

TEL: +31 (0)6 28851676
[Please leave a voicemail]

ADDRESS:
Meer en boslaan 19
2103 VM Heemstede

PROFILE

I am a full-stack web developer who loves turning ideas into user-friendly, high-performing web applications that are easy to maintain. Over the past 20 years, I have learned how important it is to write clear, testable, consistent, concise, and self-documenting code. In doing so, I try to follow the **SOLID (https://en.wikipedia.org/wiki/SOLID)** principles as much as possible. A combination of Django (Python) and React (with MobX) proves ideal in most cases. It allows me to deliver high-quality end products in the shortest possible time.

PROFESSIONAL EXPERIENCE

2010-present

Freelance web developer

Configurator - Transforming a research project into a **data-driven, user-friendly web application (/en/portfolio/#19)** for a European consortium of heating equipment manufacturers. Originally, it was a console-based prototype that imported and exported text files with data. This was rebuilt into a web application, refactored, and optimized (20-40x speed improvement). Additionally, a fuzzing component was added to automatically verify and stress-test calculations. (Django)

KeyAgency - Developing an asset management system for **Key Agency (https://keyagency.nl/)** to streamline content rollout processes for the product configurator of a leading motorcycle manufacturer. The process involved handling large quantities of images that had to be imported and reviewed to enable a product configurator where visitors could equip motorcycle models with accessories and immediately see how they would look. (Flask, React)

MateForm - Web form building platform MateForm (NodeJS/React) (own project)

SmartRollout - For **SmartRollout (https://www.smartrollout.com)**, I built a web portal and visualization components for their FileMaker-based platform.

Joboti - Creating an extensible dialog editor, integrating various NLP models and APIs and rebuilding the chatbot state machine architecture for **Joboti (http://www.joboti.com)** (now **Textkernel (https://www.textkernel.com/)**).

MataCSM - For Rick Olson, I built MataCSM, a system that provides organizations with insight into the security of their online assets. Servers and web applications can be added and are then regularly scanned for (potential) issues. Users are immediately notified of new developments. The system also checks automatically for leaked information from the organization (e.g., via public GitHub repositories). Additionally, servers can be added automatically from various cloud providers.

IIQ Quest - For **Implementation IQ (IIQ) (http://www.implementation-iq.nl)**, I implemented a web-based questionnaire package for conducting their IMPLEMENT research. It enabled entering questionnaires via the browser, creating invitations (imported from a vCard database, if necessary), sending them out, and tracking respondents' progress. Based on the results, people could be invited to participate in a blog platform. IIQ staff could easily add new content using Markdown, and invitees could comment or start discussions.

Smaakwijzer - For the designers of **ONTWARD (http://www.ontward.com)**, I implemented a questionnaire package that made it possible to gauge the taste of their (potential) clients.
In this case, the questionnaire editor was based on a custom Django admin manager. By simply defining the models, it was possible to easily create a complete questionnaire hierarchy on the same page instead of having to create separate objects in separate views in the standard Django admin.

Implementation Analyzer - For **Implementation IQ (IIQ) (http://www.implementation-iq.nl)**, I implemented a **system (https://www.healthcareinnovationportal.nl/)** to visualize the implementation of innovations. Users can input various data about stakeholders and their needs, and real-time graphs automatically indicate where the main obstacles are and how to address them (e.g., by providing scientific evidence of effectiveness). Questionnaires can also be generated and sent to stakeholders (which they can fill out directly in their browser) and are processed as additional data. It is also possible to view overviews and reports for an entire suite of innovations simultaneously.

Ontzameling - In collaboration with Edward Janssen (designer, Ontward), I developed the first version of the website for Stichting Ontzameling. The website serves as a catalog for a large art collection. Through the Django admin interface with custom widgets, the art collection can be managed, and the pieces can be assigned to the appropriate categories on the site. The website itself featured animated widgets showcasing art pieces, ensuring that there was always something new to see, tailored to the specific user. The current version of the site was created by onebite.

Els Streefkerk - Developed a **personal website (http://www.elsstreefkerk.nl)** for music teacher Els Streefkerk using Django CMS, based on a design by Edward Janssen.

Viola Groenhart - Voor filmmaakster Viola Groenhart heb ik haar persoonlijke **site (http://www.violagroenhart.com)** geïmplementeerd. De site kan worden onderhouden via Django CMS inclusief custom plugins om de verschillende widgets op de site aan te passen.' %}

Jan Beutener - Created a portfolio **website (http://www.janbeutener.nl)** for painter Jan Beutener, with content manageable via the Django admin interface.

Jouwstraat.nl - Set up a development environment and built a module for **jouwstraat.nl (http://www.jouwstraat.nl)** that allowed contacts to be imported from Gmail, Yahoo, and Hotmail.

Second Company - Developed a demo website for **Second Company (http://www.secondcompany.nl)** that integrated with a centralized authentication and data system using REST services.

de man en de vos - Implemented the first version of the website for the filmmakers' collective **'de man en de vos,' (http://www.demanendevos.nl)** featuring film-like transitions (fade-out, crossfade, etc.) created with jQuery. The content is editable through the Django admin interface with custom widgets.

Catastrophist.org - Built a **website (http://www.catastrophist.org)** for geologist Han Kloosterman as an archive for his research.

2009

Worked at the service desk of the internet application provider **BlueDome (http://www.bluedome.net)**.

2008

Performed maintenance on an SQL database environment and provided helpdesk support for the ICT department at **Coty Netherlands (http://www.coty.com/)**.

2008

Implemented framework-based websites at **Synetic (http://www.synetic.nl)**, primarily focused on building webshops using Drupal (PHP).

2007-2009

Worked as projectionist and usher at Filmhuis Griffioen.

2007

Conducted functional testing of various websites at Airtrade Holland and researched the transition to automated testing, focusing on testing and script development using Selenium and QEngine.

2006

Responsible for migrating an outdated database to a new in-house developed application at TBMA Europe.

2003-2004

System tester at TNS-NIPO, testing the functionality of a software suite for telephone interviewers, partially automated using QARun.

EDUCATION

2002-2007

The first two years of Computer Science, followed by a few months of Informatics, and since 2004 Artificial Intelligence at the Vrije Universiteit Amsterdam.

1995-2002

Atheneum at College Hageveld in Heemstede, graduated in 2002.

TECHNICAL EXPERIENCE

WEB DEVELOPMENT

At the moment, this is my preferred stack:

Django (Python) - I have been building web applications with Django for over 15 years. Although the role of Django has diminished in recent years due to the increased use of SPAs (Single Page Applications), it remains a solid foundation for most (web) projects.

ReactJS / MobX - I transitioned from jQuery to AngularJS and eventually found the right balance of control and compactness in React. I have now used React in several large projects together with MobX, which has proven to deliver highly efficient, powerful, and maintainable code.

Parcel / webpack / Babel - To use modern versions of JavaScript in the frontend and minimize the download size of code for users, I usually opt for Parcel or Babel and Webpack. I rarely build React-based applications without these tools.

PostgreSQL - Django loves Postgres (more than MySQL). Despite the rise of NoSQL solutions, PostgreSQL often proves to be the best choice, even for larger datasets.

HTML5 / CSS3 / Javascript (ES5, ES6) / JSX / TypeScript / TSX / SVG - Over the past 15 years, I have followed the development of new web standards and have made full use of the opportunities they offer.

Git / Apache / Ubuntu / Docker / virtualenv - Since I build most projects independently, this also means that I am usually responsible for setting up the infrastructure. This includes developing a deployment strategy and configuring the various machines.

Supplemented with:

Flask (Python) - For some prototypes, I used Flask instead of Django as Flask projects are much easier to set up and deploy (although they often lack default security).

FastAPI (Python) - Used for a personal project (mainly to get the hang of it), and it worked fantastically. A test-driven approach aligned very well with the functionality offered. I look forward to using it more often in the future.

JQuery - Used intensively years ago, but now only sporadically for small pieces of functionality where React would be overkill or for maintaining legacy projects.

Bootstrap - I use it in most projects during the initial phase so that I don't have to waste time on CSS to make applications look good from the start.

RabbitMQ - For more complex projects, I use RabbitMQ to enable communication between long-running background processes.

Django REST framework (DRF) (Python) - In the past, I used DRF for some projects, but now I would use FastAPI for that.

Django CMS (Django/Python) - Created websites for several clients that included this CMS.

Elasticsearch - Full-text search functionality for MataCSM

Celery (Python) - Used over the past year for executing tasks in the background.

Not (strongly) related to web development:

C / C++ / x86,x64 assembly - In my younger years, I programmed extensively in C and C++. Among other things, a 3D engine and various security tools. Later, only for extremely time- and/or memory-critical parts in my own Python projects. Nowadays, that is no longer necessary due to technologies like Cython.

TensorFlow / PyTorch / Theano / scikit-learn / OpenCV / Numpy / Scipy (Python) - {% trans 'Voor verschillende eigen projecten heb ik veel gebruik gemaakt van deze platformen. Tegenwoordig metname TensorFlow of PyTorch. Niet alleen het gebruik van bestaande modellen maar ook het opzetten van een eigen architectuur op basis van een dataset (al dan niet zelf gecreeerd).

Java - Completed an extensive three-part course using Java at the VU (Introduction to Programming 1 and 2, Data Structures).

MORE ABOUT ME

In 2010, I started my own web development company with the goal of helping people realize their ideas on the web. When I'm not working for clients, I am usually busy with my own creative projects. Currently, I spend a lot of time on film, photography, and visual effects, with a focus on creating music videos (both for my own music and that of others).