

Application at Canonical for

Web Frontend Engineer - JS, CSS, React, Flutter

Sergei Ivanov

IT Süsteemide arendus, Tartu Ülikooli

Hiring lead: Anthony Dillon

November 9, 2025

Abstract

This document provides my written interview responses for Canonical's assessment stage. It outlines my web and software engineering experience, education background, and personal reflections on Canonical's mission and goals. Over the past year, I have deepened my skills across Docker, Rust, Next.js, and deployment pipelines, gaining valuable insights into microservice design, automation, and DevOps workflows.

The following sections summarize my practical experience in full-stack development, backend infrastructure, and teamwork in both academic and professional environments. I also discuss my perspectives on open-source collaboration, software performance, and quality practices. Finally, I reflect on Canonical's purpose within the open-source ecosystem, identifying what inspires me about its mission and how I hope to contribute to its continued growth and innovation.

Keywords: JavaScript, React, Rust, NextJS, Skills, Infrastructure, Context, Education, Python, Machine Learning, Docker

Contents

Overview	2
Questions	3
Web engineering experience	3
Software engineering experience	3
Education	3
Context	4
Answers	5
Web engineering experience	5
Software engineering experience	6
Education	7
Context	7

Overview

For the reference, I would like to present **Assessment description**:

1. The Written Interview: As the first part of this stage we invite you to submit your answers to some questions specifically designed so that we can find out more about your experience and skill set. This is your opportunity to highlight your key strengths for this position. To ensure that we mitigate against unconscious bias in our hiring process we ask that you refrain from including any personal identifiable information in this document.
2. Psychometric assessment: As part of our professional review this assessment will help us to understand your fluid intelligence.
3. Technical tests: Some roles will require a technical assignment to be completed.

There are several parts to this stage: a written interview, a psychometric assessment and some roles will require a technical assignment to be completed.

Please address the following in a PDF document of no specific word count. Feel free to reuse answers if you have made applications to multiple roles at Canonical.

Web engineering experience

Questions

Also, I present all questions (that are asked to me) here:

Web engineering experience

1. What skill or knowledge have you acquired in the past year that has been particularly helpful?
2. What motivated you to learn it?
3. What has the impact been for you and your team?
4. Describe your experience of web programming - JavaScript, Typescript, React, CSS and Python in particular.
5. Describe your experience building large systems with many services - web front ends, REST APIs, data stores, event processing and other kinds of integration between components.
6. What are the key things to think about in regard to architecture, maintainability, and reliability in these large systems?
7. Describe any experience you have with Flutter.

Software engineering experience

1. What kinds of software projects have you worked on before?
2. Which operating systems, development environments, languages, databases and frameworks?
3. Outline your thoughts on open-source software development. Have you been an open-source maintainer, can you describe the scope of your contributions to those projects?
4. What software products have you yourself led which shipped many releases to multiple customers? What was your role?
5. What is your most senior role in a software engineering organisation?
6. Describe your span of control and the diversity of products, functions and teams you led.
7. Outline your views on the role of an engineering manager in shaping a high-functioning team.
8. Outline your thoughts on performance in software engineering. How do you ensure that your product is fast?
9. Outline your thoughts on quality in software development. What practices are most effective in software teams to drive improvements in quality?

Education

1. We consider academic results in high school and university for all roles, regardless of seniority. In every discipline, from engineering to marketing to operations and sales, we intensely value colleagues who are able to puzzle through difficult problems and find the optimal path forward.
2. How did you rank in your final year of high school in mathematics? Were you a top student? On what basis would you say that?
3. How did you rank in your final year of high school, in your home language? Were you a top student? On what basis would you say that?
4. Please state your high school graduation results or university entrance results, and explain the grading system used. For example, in the US, you might give your SAT or ACT scores. In Germany, you might give your scores out of a grading system of 1-5, with 1 being the best.
5. Can you make a case that you are in the top 5% in your academic year, or top 1%, or even higher? If so please outline that case. Make reference where possible to standardised testing results at regional or national level, or university entrance results. Please explain any specific grading system used.
6. What sort of high school student were you? Outside of class, what were your interests and hobbies? What would your high school peers remember you for?
7. Which university and degree did you choose? What other universities did you consider, and why did you select that one?
8. Overall, what was your degree result and how did that reflect on your ability? Please help us understand the grading system for your results.
9. During all of your education years, from high school to university, can you describe any achievements that were truly exceptional?

10. What leadership roles did you take on during your education? Did you conceive of, and drive to completion, any initiatives outside of your required classwork?

Context

1. Outline your thoughts on the mission of Canonical. What is it about the company's purpose and goals which is most appealing to you? What is risky or unappealing? Are there any elements of the company goals that you are unsure about?
2. Who are competitors to Canonical, and what does Canonical need to change to be a more effective competitor?
3. What would you most want to change about Canonical?
4. What gets you most excited about this role?

Answers

Web engineering experience

During this year, I acquired many skills; of them, the most vital was Docker. While I failed to deliver aesthetic beauty of the client-side application (*NextJS*), I highly focused on Docker + SpringBoot which opened new roads for me - deployment, devops, etc. As a side bonus, I deployed SpringBoot to *AWS* and learned about real deployment, which revealed the natural microservices architecture. Months later, in September, I had to defend Windows Administration course in college and to do that, I learned:

- How to mount my disk (net use command on Windows);
- How to use EduVPN to access college network correctly and grab the ISO files
- How to use Virtual Machine (VM and Oracle Virtual Box)
- Router manipulation: Port forwarding and NAT
- The usage of CloudDNS followed up by Google's DNS experimentation to ping and trace my own Rust server

These are important steps of end-to-end deployment. The greatest motivation was that I could not fail the course due to graduating in next year. If I were to fail, I would have to study one year more and that would ruin my graduation. I had to re-use my existing knowledge and adapted fast to Windows course.

You can watch the defense of my labs here:

- Lab 1
- Lab 3 & Lab 4

Please, be certain that these videos are in Russian and English as my professor's mother tongue is Russian.

Due to University requirements, I had to use both languages. If professor insisted on Estonian, I would use that one; yet, professor never did.

During the 2025 year, I had to work in team for *tomodachi* project which you can find NeXus Central and navigate to *tomodachi_core*. The most difficult was to mentor teammates as their proficiency in Python was relatively low. They used

```
class Person(object): ...
```

syntax which is obsolete in Python 3.14, in 2025. Python has grown so much as to adapt to the new features. That is why the biggest impact for me was teaching, mentoring, collaborating and discussing the work done so that we could defend it. The impact for my team was gaining new insights into Rust, Python and slightly into microservices, docker worlds. The team lead delegated tasks well; however, nobody used Jira for documentation (documenting task, commenting on progress and then documenting merge request). Of my team, I was the one responsible for Git commands and Version Control of our product. I believe that they did great for the reason of our work defended successfully (exam!).

JavaScript was my first programming language. It was my first passion after C++. I learned the syntax and, most importantly, DOM manipulation so as to make my website interactive with JS. Of the simplest things, I studied loading animation, background animations (this leads to THREE.js and C++ powered Shaders), Drop down menu and etc. Later I switched to NextJS as it was the React framework (NextJS is built on top of React!) and brought SSR, SSG benefits. It led to use TSX to build pages and add Framer Motion to animate elements. The most important benefit was re-usability of components and API. In 2025, I used NextJS with Rust's Axum and I liked it so much. With NextJS's framework, I had access to Server Actions and could leverage more efficient way of handling the form.

CSS, however, is the my weakest point. Sure, I do understand a diverse number of concepts and was self-taught to study advanced CSS features (animations, keyframes, measures (1 rem, 1 em, 1 vh, etc)). I use TailwindCSS for writing CSS because it is simple and efficient. I can refer to both:

- AI (Gemini, ChatGPT, Microsoft Copilot)
- Documentation

This helps me achieve the basic appearance of webpage and lets me focus on client-side programming that tied to the back-end or users. That is, sometimes client-side is not always communicating with back-end to do one action. Sometimes it can act on its own to deliver product to the user.

Python was the game-changing programming language, since it was the simplest to learn and use. Python's main advantage is said to be accessibility. There are so many libraries that one can re-use an existing solution rather than waste entire month for their own. I used Python for scripting, back-end and lately for ML, due to college Data Science course.

The 5th question can be answered with my gRPC, Rest trials implemented in Rust. I used Rust for the server-side programming and TypeScript (NextJS + Deno) for the client-side. The simplest API was REST as Rust's Axum crate is the easiest to start with. This leads to microservices: one can create as many services as they need to better organize the code (one part focuses on one requirement and another for the other one). For database, I had experience of using Postgres and sqlx crate in Rust. I also used MongoDB in JS and Python. Due to the database connection logic remaining relatively the same, this does not stop me from using Postgres in TS (In fact, I did it with Prisma). The Prisma was the best experience that I have ever had. It was the easiest way to work with SQLite and PostgreSQL databases.

For deployment, I used Docker; my latest task is Discord Bot for Discord and then Rust's service (Axum) written for Jira (and uses jirrust crate). This demonstrates that while being intern, I had to think like a tiny senior and approach my task with the concern about optimization, efficiency, security and architecture (token hiding, traffic handling so company does not get overpriced for the use of token). I also read about infrastructure and possess basic skills for that.

Related to Flutter, I can only leverage my Kotlin and Android development knowledge. All I remember about Flutter is that it is the most reliable framework for Android platform app development using React. In other words, given my fast adaption, I can learn Flutter while leveraging my other skills (Rust, TS, React, Kotlin, etc).

Software engineering experience

I worked on many software projects before. My most favourite is Python's one where I had to learn Machine Learning, then gRPC Rust example when I used TLS 1.3 and gRPC API to implement secure service. I also enjoyed Axum + sqlx so much in the context of server-side of a larger system. I even implemented basic console app written in C# (2024, December) which you can find in NeXus Central (the aforementioned link) and navigating to tomodachi_games.

Throughout my learning and practically programming, I used Linux, Windows, VM (Virtual Machines), and Docker. For development environment and version control, I used GitHub, Jira and VIM for coding. Practically, I mostly use git bash and vscode. Languages are listed on my CV: Kotlin, JS/TS, Python, C#, Rust, Lua, Zig, Java. Finally, databases and frameworks: for the databases, the list is MongoDB, PostgreSQL, SQLite; for the frameworks, it is React, NextJS, SpringBoot.

The rest questions, I will answer in the list due to time constraint:

Outline your thoughts on open-source software development. Have you been an open-source maintainer, can you describe the scope of your contributions to those projects? No, I did not yet; I mostly participated in the informational sense: read the code, studies patterns;

What software products have you yourself led which shipped many releases to multiple customers? What was your role? In real companies, I did not; but in educational, I was the leading engineer and my role was to mentor, test and deliver.

What is your most senior role in a software engineering organisation? Intern.

Describe your span of control and the diversity of products, functions and teams you led. From websites, console application, desktop application and machine learning.

Outline your views on the role of an engineering manager in shaping a high-functioning team. My view is very simple: engineer must read requirements well, talk professionally to their supervisor, understand users, stakeholders and then implement the system required for the product, while delegating tasks within their team.

Outline your thoughts on performance in software engineering. How do you ensure that your product is fast? Mostly, the libraries already cover huge performance benefits. For example, JavaScript while slow, can be optimized with C++ NodeJS addons. Or you can use Zig/Rust and gRPC to speed up. However, practically, GraphQL is the best, and Rust supports it. Rust is the blazing fast and leads to efficient product. To test, we use libraries like pytest in Python to bench and run performance checks on the code. We can use VTune by Intel and inspect further for bottlenecks.

Outline your thoughts on quality in software development. What practices are most effective in software teams to drive improvements in quality? I believe they use meetings quite often to discuss what is being done, how to do and etc. For example, in my current company (internship - unpaid), I listen to different departments, their deliverables, and how to proceed (next plans); however, when it comes to Jira, it is mostly agile or kanban software development strategy.

Education

For Education, I sum up pretty quickly:

- Bachelor degree;
- Grade is B (average or 4.0 and up)
- Major: IT System Development
- Dedication is very high
- Wrote my own book (available on Amazon)
- Possess highly philosophical, physics, mathematics and chemistry skills. Also got strong Biology reasoning due to physics, mathematics reasoning.
- Read Landau book about least action principle.

Context

Outline your thoughts on the mission of Canonical. What is it about the company's purpose and goals which is most appealing to you? What is risky or unappealing? Are there any elements of the company goals that you are unsure about?

Mission of Canonical Canonical's mission is to make open-source software accessible, secure, and reliable at scale; particularly through Ubuntu and its ecosystem. What appeals most to me is how Canonical promotes open-source innovation while ensuring enterprise-level stability. It blends freedom and professionalism, enabling individuals and organizations to build confidently on open technologies.

The main risk is that Canonical's business model heavily depends on community trust and long-term support contracts; any drift from community alignment could weaken its foundation. I am also slightly unsure about how Canonical balances its strict remote-work structure with collaboration and innovation speed.

Who are competitors to Canonical, and what does Canonical need to change to be a more effective competitor?

Competitors & What Canonical Could Improve Key competitors include Red Hat (IBM), SUSE, and AWS/Linux-based cloud services. Canonical stands out for its lightweight, developer-friendly systems, but could strengthen its position by:

1. Investing more in developer experience tooling and community engagement.
2. Expanding integration partnerships with cloud and AI platforms.
3. Simplifying its enterprise onboarding and documentation for smoother adoption.

What would you most want to change about Canonical?

What I Would Change I would improve communication and visibility between Canonical's teams and the open-source community; more transparency, public roadmaps, and mentorship-style contributions could boost community involvement and product alignment.

What gets you most excited about this role?

What Excites Me Most About the Role I am most excited about contributing to products that power millions of servers, IoT devices, and developer workflows worldwide. Working in a fully remote, highly technical environment that values both efficiency and open-source principles perfectly fits my interest in scalable infrastructure, automation, and modern DevOps practices.