

ADITYA SHANKAR

FULL-STACK-ENTWICKLER · BERLIN, DEUTSCHLAND

me@river.berlin / river.berlin / GitHub / LinkedIn / Blog (RunPod)

Englisch und Hindi als Muttersprachen · Deutsch — Ziel: C1-Zertifikat bis Dezember

DARF ICH MICH VORSTELLEN?

Ich bin Aditya Shankar, Full-Stack-Entwickler mit rund 7 Jahren Erfahrung und wohnhaft in Berlin. Ich arbeite mit TypeScript und Python in Frontend, Backend und GPU-Infrastruktur. Zuletzt war ich Softwareentwickler bei [RunPod](#), einem Cloud-Anbieter mit Spezialisierung auf KI-Dienste, wo ich einer der ersten eingestellten Entwickler war und ein Team von vier Personen geleitet habe. Mein Schwerpunkt lag auf dem Fine-Tuning und Deployment von LLMs und Vision-Language-Action-Modellen, der Optimierung von S3-Verbindungen, Netzwerkspeicher und Hugging-Face-Modell-Caching sowie dem Verfassen von Blogartikeln und Dokumentation.

In meiner Freizeit beschäftige ich mich mit Robotik und Three.js. Ich habe Simulationssysteme für inverse Kinematik gebaut ([robots.river.berlin](#)) sowie Mechanismen, um Objekte mit ArUco-Markern zu erkennen und aufzunehmen — auf Basis des SO-101 von LeRobot ([instant.river.berlin](#), [GitHub](#)).

BERUFSERFAHRUNG

RunPod, Softwareentwickler

2023–2025

Vollzeit (40 Std./Woche) 2023–2024, danach 20 Std./Woche neben meinem B.Sc. · [runpod.io](#)

- Ein Cloud-Anbieter mit Spezialisierung auf KI-Dienste, inzwischen mit 1 Mrd. \$ bewertet und mit über 1 Mio. Kundinnen und Kunden auf der Plattform. Ich war einer der ersten eingestellten Entwickler und habe ein Team von vier Personen geleitet.
- Fine-Tuning und Deployment großer Sprachmodelle und Diffusionsmodelle; Beratung von 1.200 Kundinnen, Kunden und Organisationen zu Deployment, Optimierung und Produktivsetzung GPU-basierter VM-Deployments.
- Arbeit mit PyTorch und Hugging Face Transformers sowie mit nativem CUDA-Code zur Optimierung des Modell-Deployments.
- Optimierung von S3-Verbindungen, Netzwerkspeicher und Hugging-Face-Modell-Caching über Tausende von Maschinen hinweg.
- Verfassen zahlreicher Blogartikel für RunPod ([hier verfügbar](#)) sowie großer Teile der Entwicklerdokumentation für RunPods Python-Bibliothek.

Repunzel, Full-Stack-Entwickler

2020–2023

Vollzeit (40 Std./Woche) · [repunzel.com](#)

- Aufbau einer Bewerbungs-Pipeline, mit der Schauspielerinnen und Schauspieler ihre Portfolios sofort an über 1.000 Agenturen senden können — mit deutlich besseren Chancen auf ein Engagement.
- Arbeit an Frontend und Backend mit Svelte, TypeScript, Python, Flask, Postgres und Firebase.
- Entwicklung von CI/CD-Pipelines für eine ausfallsichere, robuste Architektur; Automatisierung von E-Mail-Validierung, Zustellungs-Tracking und Testumgebungen.
- Aufbau automatisierter E-Mail-Tests und -Auswertungen, wodurch 8 Stunden manuelle Arbeit pro Woche auf eine 30-sekündige Bewerbung reduziert wurden.
- Entwurf einer Architektur zur zuverlässigen Planung und Stornierung von Aufgaben mit RabbitMQ und Python.

TypeCodeLearn, Full-Stack-Entwickler & Tutor

2019–2020

Auf Vertragsbasis, freiberuflich

- Unterricht in Data Science, NumPy, Pandas, yfinance und Python, um Studierende auf automatisierte Trading-Anwendungen zu spezialisieren.
- Ersetzung der gesamten manuellen Zahlungsabwicklung durch automatisch dokumentierte Online-Zahlungen über Stripe.
- Verwaltung von OAuth, E-Mail, SMS, SendGrid, Twilio, Stripe und Google Ads.
- Erstellung und Pflege von Python-Kursinhalten.

Freiberuflich, Softwareentwickler

2019–2022

Auf Vertragsbasis · [upwork.com](#)

- Erstellung und Wartung von Websites für über 20 Auftraggeber.
- Scraping und Analyse von über 20.000 Immobilientransaktionen zur Identifikation optimaler Angebote — Wochen manueller Arbeit wurden auf Minuten reduziert.

Studentische Hilfskraft in der Lehre, Arizona State University · 2 Semester

PROJEKTE

Die-Roboter [robots.river.berlin](#) · [GitHub](#)

Browserbasierte Physiksimulation für LeRobot-Arme. Three.js/enable3d-Szene, Laden von URDF-Modellen, ammo.js-Physik, pivotbasierte Steuerungszuordnung. In TypeScript geschrieben.

SO-101 Pick-and-Place [instant.river.berlin](#) · [GitHub](#)

ArUco-Erkennung, Homographie, MuJoCo, inverse Kinematik im geschlossenen Regelkreis. Teilweise eine KI-gestützte Portierung des Thesis-Codes nach JavaScript, durchgehend geprüft und korrigiert; die native serielle Schnittstelle wurde durch WebSerial ersetzt, um den Roboter direkt im Browser zu steuern. Vorgestellt in der c-base, Berlin.

Bachelorarbeit eingereicht im Juli 2026

Empirischer Vergleich klassischer inverser Kinematik mit einem Vision-Language-Action-Modell für robotisches Pick-and-Place an einem SO-101-Arm.

Außerdem: [build-docker-in-runpod](#), [flask-value-checker](#) (auf PyPI veröffentlicht), [leassemblee](#) & [lerobot-training-api](#) (LeRobot-Tooling), [modulify](#)

OPEN SOURCE & WEBSTANDARDS

Mitarbeit an der Firefox-Entwicklung, unter anderem Verbesserung der Mozilla-Dokumentation (zum Beispiel [hier](#)), Teilnahme an der Konferenz der CSS Working Group zum Austausch mit Browser-Entwicklerinnen und -Entwicklern, Beiträge zur Verbesserung der Barrierefreiheit und zur Standardisierung von Formular-Layouts in Browsern.

HACKATHONS

Gewinner, Chaos Unicorn Hacks [c-base, Berlin](#) · Oktober 2025 · [Event-Link](#)

DWebCamp Hacks [c-base, Berlin](#) · Juli 2026 · [Event-Link](#)

Gewinner, CodeDevils Annual Hackathon [Arizona State University](#) · 2020

AUSBILDUNG

IU Internationale Hochschule, B.Sc. Angewandte Künstliche Intelligenz 2022–2026 (vorauss.)

Arizona State University, Software Engineering (Studienleistungen) 2018–2020

SPRACHEN & KENNTNISSE

BACKEND

Linux, Python, C, Shell-Skripte, Django, Flask, NodeJS, NextJS, JavaScript, TypeScript, PostgreSQL, GCP

DEVOPS

AWS, Docker, Docker-Compose, Cloud Functions, Cloud SQL, Cloudflare, DigitalOcean, Git, pytest, unittest, Mocha.js, Puppeteer

FRONTEND

HTML, CSS, TypeScript, React & React Native, NextJS, Vue, Svelte, Three.js/WebGL

MASCHINELLES LERNEN

PyTorch, TensorFlow, vLLM, CUDA, OpenCV, Hugging Face Transformers