RODION MAYATSKIY

Birmingham, Alabama
mayatrod@gmail.com

Portfolio: rmaytr.github.io
LinkedIn: linkedin.com/in/rodion-mayatskiy-025128374
GitHub: github.com/rmaytr

EDUCATION

University of Alabama at Birmingham | Honors College · College of Arts & Sciences — Aug 2025 – expected May 2029
B.S. Computer Science & B.S. Physics (double major) · Concentration: Data-Driven Modeling and Technology · Minors: Mathematics, Entrepreneurship

- **GPA 4.0 · Presidential Honors List** (Fall 2025, Spring 2026)
- **Early admission** to UAB's Accelerated Bachelor's-to-Master's Program in Artificial Intelligence

Completed: CS 103 Introduction to CS in Python · HC 100 Honors FYE · HC 118 HonSem: Fantastic Film · MA 126 Calculus II · PH 221 General Physics I · CS 203 Object-Oriented Programming · CS 250 Discrete Structures · MA 227 Calculus III · PH 103 The World Through Data · PH 222 General Physics II

IB transfer credit: MA 125 Calculus I · PH 201 College Physics I · PH 202 College Physics II

In progress (Fall 2026): CS 113 AI in a Nutshell · CS 303 Algorithms and Data Structures · CS 380 Mathematical Foundations of AI · MA 252 Intro Diff Equations (HON) · MA 360 Scientific Programming · PH 104 Community Data Research · PH 110 Topics in Contemporary Physics

Rancho Solano Preparatory School, Scottsdale, AZ — Aug 2023 – May 2025

- **IB curriculum** (HL Physics, HL Mathematics), on scholarship
- **GPA 4.0 unweighted** / 4.47 weighted · **SAT 1550**

RESEARCH & PUBLICATIONS

Undergraduate Researcher, Computational Plasma Physics | Computational plasma physics research group, UAB Department of Physics, Birmingham, AL — May 2026 – Present

- Write and run coupled **laser-ablation and fluid-dynamics simulations** of laser-generated plasmas in **C** with the **Basilisk** PDE framework.
- Run workloads on UAB's **Cheaha supercomputer**; learned **Linux and Bash** from scratch to manage HPC jobs, environments and data.
- Work with **synthetic plasma datasets** in which plasma temperature (T) and ionization fraction (Xe) serve as ground truth for testing ML models.
- Write **Python** scripts that process, visualize and analyze simulation output into the plots and comparisons the team uses to evaluate models.
- Support research on **ML-guided pulsed laser deposition (PLD)**, where **Gaussian Process Bayesian Optimization** builds surrogate models of plasma conditions from a small number of experiments.

Tools: C · Basilisk · Python · Bash · Linux · HPC (Cheaha) · Gaussian Process Bayesian Optimization · Scientific visualization

"Machine Learning-Assisted Pulsed Laser Deposition Based on Plasma Optical Emission Signatures" — Nov 2026
D. Carpenter, Z. Nasiri, S. Gupta, **R. Mayatskiy**, S. B. Harris (Center for Nanophase Materials Sciences, Oak Ridge National Laboratory), R. P. Camata

- **Co-author** of an abstract accepted for an **oral presentation** at the **AVS 72nd International Symposium**, Pittsburgh, Nov 8–13, 2026.
- Across **1,000** independent trials, the model found optimal plasma conditions in as few as **three** adaptive iterations: a step toward autonomous, real-time control of thin-film growth.

WORK & TEACHING EXPERIENCE

Teaching Assistant, CS 103: Introduction to Python | UAB Department of Computer Science, Birmingham, AL — Aug 2026 – Present

- Appointed **primary TA** by the instructor for a **60+ student** course (~10 hrs/week).
- Grade **all homework and lab assignments** as the course's **only grader**, keeping feedback consistent for every student.
- Teach my own weekly **2-hour lab section of ~40 students**, walking through hands-on exercises from first principles.
- Hold **2 hours of office hours** a week on programming concepts, problem-solving and students' own code.
- Work with a second TA, who leads the smaller lab section, on day-to-day course operations.
- Built a small **recursion visualizer** in Python to explain recursion to my students.

Skills: Python · Teaching · Code review · Technical communication

Web Developer Intern | BOR Realty, LLC, Phoenix, AZ Summer 2026

azrussianrealtor.com

- Built a **bilingual (English/Russian)** website in **Next.js 14, React, TypeScript and Tailwind CSS**, published as **18 static pages**, replacing the brokerage's main site.
- SEO and AI-search work: hreflang tags, XML sitemap, Open Graph/Twitter cards, **schema.org JSON-LD** (RealEstateAgent, FAQPage, Review, AggregateRating).
- **Automated media library: YouTube Data API v3** with an RSS fallback, a **Node.js** sync script and a **daily GitHub Actions** job keep **689 videos** current with no manual work. Videos are auto-categorized by hashtag and title, duplicate Shorts are hidden, and the library supports filtering, date sorting and pagination.
- Sold-listings and testimonials built on the same **typed data model**, so they can be synced the same way.
- **WCAG 2.1 AA: 0 axe-core violations** across 40 page audits; **300+ Playwright tests** in Chrome and WebKit from 320 to 1440 px.
- Re-encoded the hero video with **ffmpeg** from 37 MB to 8.9 MB (**76% smaller**), plus a 1.9 MB vertical cut for phones.
- Wrote the **privacy, accessibility and fair housing** pages following ADA, CalOPPA and Arizona real estate advertising rules.

Stack: Next.js 14 · React · TypeScript · Tailwind CSS · Node.js · YouTube Data API v3 · GitHub Actions · Playwright · axe-core · ffmpeg · Vercel

Academic Peer Tutor | UAB Vulcan Materials Academic Success Center, Birmingham, AL Feb 2026 – May 2026

- Tutored students across **12+ STEM courses**, including Calculus I/II, General Physics and Python (~5–6 hrs/week).
- The center's **only General Physics I tutor**.
- Used **inquiry-based problem solving**, graphing and online simulations to help students build intuition.
- Taught **TI-Nspire CX II** use for calculus: verifying solutions, writing custom functions, graphing 3D surfaces.
- Moved into the **CS 103 primary TA** role in Fall 2026.

Skills: Calculus · Physics · Python

Web Content Editor Student Assistant (Front-End Web Developer) | UAB Student Affairs Marketing & Communications (SAMAC), Birmingham, AL Oct 2025 – Present

- Maintain and update **1,500+ pages** across **20+ UAB department sites** in **Joomla 5**, writing custom HTML, CSS, JavaScript and Bootstrap.
- Converted a **40-page PDF annual report** into an **interactive, animated webpage** as part of a **3-developer team**.
- Asked to update a department's video library page, delivered a **full redesign**: content reorganized and categorized, a cleaner accessible layout and a clear information hierarchy.
- Run ongoing **digital accessibility reviews** (WCAG, WAVE) of UAB pages, starting with Dining.
- Started in my first semester, at up to **18 hrs/week** alongside a full course load.

Skills: Joomla 5 · HTML · CSS · JavaScript · Bootstrap · WCAG/WAVE

PROJECTS

HackBama_01 | 1st place in the EcoCAR track Oct 17, 2026 · 12-hour hackathon at the University of Alabama

Racing-line planner, physics-based prediction, ECO mode, visualizer, team coordination · Team: HackUAB (Rodion, Saisi, Viraj)

github.com/rmaytr/HackUAB · youtu.be/a-963WiK0cQ · uab-track-sync.base44.app

- **Safe corridor:** turns the 40 m preview into a corridor shrunk for the car's width, reshaped around obstacle cones (picks the wider gap, with hysteresis).
- **Racing-line planner:** a **minimum-curvature racing line** re-solved every step as a regularized least-squares problem in pure **NumPy**, warm-started for stability; a grip-based speed profile with forward/backward passes and a **visibility guard** for sharp corners just past the horizon.
- **Physics-based prediction & plan verification:** a vectorized **dynamic bicycle model** of the AWD car (tyre saturation, load transfer, 40/60 drive split, steering lag, drag) simulates the next **1.5 s**, and slows the plan wherever predicted grip or rear-tyre slip would exceed limits.
- **Tracking controller:** curvature feedforward plus pure-pursuit feedback with latency compensation, and an AWD per-axle traction model (traction control/ABS-like) with online slip adaptation.
- **Safety layer:** per-stage fallbacks, timing and a compute-budget guard; a blocked-road safety stop.
- **PERFORMANCE vs ECO modes:** a built-in energy meter (energy balance within 5%) and a Pareto front of lap time vs energy. ECO uses about **30% less energy for about 10% more lap time, still with zero cones**.
- **0 cone hits, 0 off-course, 0 missed gates, 0 controller errors** across all public courses and randomly generated unseen tracks, in both modes (22 validation laps).
- About **10 ms per decision** against a 50 ms budget. **129 automated tests**.

- Owned the **racing-line planner** and the **physics-based prediction and verification** model.
- Built **ECO mode**, the energy meter and Pareto analysis, and the **blocked-road safety fix**.
- Built the **interactive visualizer**.
- **Coordinated the team:** project plan, merges and integration testing, submission, pitch and demo video.

Stack: Python · NumPy · CARLA 0.9.16 · Vehicle dynamics · Least-squares optimization · pytest · Git/GitHub · Base44 · Claude Code · OpenAI Codex

Cairn | Live beta Oct 2026 – Present
Solo product engineer · AI-assisted build with Claude Code
cairn-tasks-tawny.vercel.app · github.com/rmaytr/Cairn_Task_Tracking_App

- An installable, offline-capable **PWA** in **React 19, TypeScript, Zustand and Zod** (~**12,100 lines**) on **IndexedDB**. Everything stays in the browser unless the user turns on sync or the assistant.
- A natural-language capture box: a tokenizing parser turns "Finish lab report by Friday !" into a task with an area, a due date and a priority (**15+** date forms, DST-safe). Importers handle **5** formats (plain text, Markdown/Obsidian, JSON, Todoist CSV), each with a reviewable preview.
- A dependency-aware planner that combines urgency scoring with **Kahn's topological sort** on a priority heap and checks every new link for cycles. It orders **500** tasks with dependencies in about **10 ms**.
- **Lock in**, a focus mode that shows one task and its next small step and says why it comes first. Three views of everything: List, a mind **Map** and a **Radar** where nearer the middle means sooner.
- An optional **Claude** assistant that turns a sentence into a task with ordered steps, with **Zod**-validated structured outputs, prompt caching and prompt-injection safeguards. It falls back to local planning on any failure. Switching from Opus to **Haiku** cut estimated cost **40×** (≈ **$0.001** per task), and lazy-loading the SDK cut the main bundle **33%** (968 → 645 KB).
- Optional end-to-end-encrypted sync (implemented and tested, **not yet enabled in production**): **PBKDF2-SHA-256 (600,000 iterations)** and **AES-GCM-256** with non-extractable WebCrypto keys, ciphertext bound to its owner and version, and rollback protection, on **Supabase Postgres** with **Row Level Security**.
- A **Cloudflare Worker** proxy for shared AI access, with an origin allowlist, constant-time invite-code checks, **Workers KV** rate limits and streamed body caps, covered by **48** tests.
- A strict Content Security Policy; the build refuses to ship a secret key. The assistant sends only the new task's text plus open-task titles, areas, dates and priorities. Notes and steps never leave the device.
- Led **4** redesign rounds driven by user feedback, ending in a token-based design system with accessible menus, drag and drop, phone swipe gestures, full keyboard support and **4.5:1** text contrast. Automated design-QA failures fell from **4 to 0**.
- **846** Vitest unit and integration tests (including the proxy's 48) and **33** Playwright end-to-end tests (desktop, phone, offline). Triaged and fixed **31** findings from **3** parallel code-review passes (security, UI, logic).
- Deployed as a link-only beta on **Vercel** with a shared assistant; CI on **GitHub Actions** type-checks, lints, tests and builds on every push.

Stack: React 19 · TypeScript (strict) · Zustand · Zod · Vite · Workbox (PWA) · IndexedDB · WebCrypto · Supabase (Postgres · RLS) · Cloudflare Workers + KV · Anthropic Claude API · Vitest · Playwright · GitHub Actions · Vercel

Canvas Assignment Tracker | Private beta (6 users at several universities) Oct 2026 – Present
Solo developer · AI-assisted build with Claude Code
github.com/rmaytr/CanvasAssignmentTrackingBot

- A multi-user Telegram bot in **Python 3.13 (asyncio)** (**5,500+ lines**, **51** modules). It reads Canvas To Do data through the **Canvas REST API**, sends a ranked morning list, and while anything is due within 24 hours sends one combined reminder every 2 hours until Canvas shows it submitted. One-tap Done, Snooze, Open and Undo buttons; quiet hours; an 11:30 PM warning about anything due overnight.
- A transparent **100-point** priority score built from urgency, each item's actual share of the final grade (from weighted assignment groups) and estimated effort, with a one-line reason shown for every rank.
- A tenant-isolated architecture: a separate **SQLite** database, Canvas client and scheduler per user, with invite-only access, **Fernet**-encrypted Canvas tokens at rest and a resumable in-chat sign-up that takes about **2 minutes**.
- A pure, deterministic, DST-correct scheduler whose send keys are claimed atomically before delivery, so restarts and crashes can't produce duplicate reminders.
- Re-architected the bot from a single-user, self-hosted tool into an invite-only multi-tenant service after testers found self-hosting too complex, keeping the tested core unchanged (v0.1 Oct 17 → v0.2 "Friends mode" Oct 24, 2026).
- Canvas client is read-only (GET only), host-pinned, uses **ETag/304** conditional polling and rate-limit backoff with jitter, and runs SSRF checks on user-supplied Canvas addresses. The bot deletes any chat message containing a token, and secrets are redacted from every log line.

- Hardened **Docker** deployment (non-root, read-only filesystem, dropped capabilities, file secrets) with a one-command installer, on **Google Cloud Compute Engine**'s free tier (**$0** during the beta).
- **426** automated tests: **94%** line and branch coverage, **98%** on the scheduling core; **0** mypy --strict errors; Ruff security linting; CI on **GitHub Actions**.
- Validated by a **3-week** minute-by-minute simulation (~**28,800** ticks) across a daylight-saving change, plus **Hypothesis** property tests.
- Fixed **18** Critical or Important findings from **2** independent code reviews, each with a regression test.

Stack: Python 3.13 (asyncio) · python-telegram-bot · Canvas LMS REST API · httpx · SQLite · Pydantic · cryptography (Fernet) · Docker · Google Cloud Compute Engine · GitHub Actions · pytest · Hypothesis · mypy --strict · Ruff · uv

Pigeon Problems Oct 2026

Creator · AI-assisted build with OpenAI Codex

pigeon-problems.vercel.app · github.com/rmaytr/Pigeon-Problems

- A **Three.js/WebGL** client with a **Cloudflare Workers** backend that runs each room as its own **Durable Object** and persists room state. A 1–4 player game playable from a link, no account or install.
- Server-authoritative netcode: a fixed **60 Hz** simulation, **20 Hz** state snapshots and **30 Hz** client input, with client-side prediction and sequence-based reconciliation. Movement and checkpoint recovery verified under **300 ms** of injected round-trip latency.
- One JavaScript physics and level engine shared across the browser, a local Node server and Cloudflare, including a custom constrained-load solver for cargo carried by one or two players, and a throw preview that predicts impact up to **~3 s** ahead using the same rules as gameplay.
- Hardened sessions: secret reconnect tickets kept out of broadcasts, input-sequence and duplicate-action checks, rate and payload limits, and a **2-minute** reserved seat on disconnect. Optional **WebRTC** voice with room-scoped signaling.
- Geometry batching kept local 1440×900 Chrome samples at **~60 fps** in **100–141 draw calls**.
- Keyboard and mouse with remappable keys, touch controls and the standard **Gamepad API**, plus reduced-motion settings and HUD layout checks down to a **320 px** screen.
- **62** automated engine and session tests, **65** scripted **Playwright** browser checks, and complete runs of all 6 courses with **2 and 4** real WebSocket clients on the public deployment.
- Client and real-time backend deploy independently, with saved release versions and a documented rollback path.

Stack: JavaScript (ES modules) · Three.js · WebGL · Node.js · WebSockets · Cloudflare Workers · Durable Objects · WebRTC · Web Audio API · Vite · Playwright

sofiakamal.art May 2026 – Present

Designer & developer (independent client project)

sofiakamal.art

- Presents **58 works across 5 mediums** with **masonry galleries**, a **zoomable lightbox** (keyboard, swipe, a shareable link per artwork) and **scroll animations designed for each medium**.
- Signature **"watercolor bloom" page transition**: pigment spreads from the click point and the next page appears inside it. Built with the cross-document **View Transitions API** and **Web Animations API**, with a CSS fallback for Firefox; **zero dropped transitions** across 90 automated navigations.
- Rebuilt so pages are pre-rendered to static HTML with React via a custom **Node.js + esbuild** build: JavaScript **~1,060 KB → 58 KB**, mobile Lighthouse performance **32–41 → 87–95**, first paint **~10 s → 1.1 s**.
- Removed scroll and idle lag by replacing continuously animated SVG filters with GPU-composited layers and rewriting the custom cursor to bypass React re-renders; idle layout recalculations **303 → 2** per 5 seconds.
- **100/100 Lighthouse SEO, accessibility and best practices**: WCAG AA contrast, semantic landmarks, a focus-trapped keyboard-accessible lightbox, alt text for every artwork, reduced-motion support.
- Technical SEO and AI-search optimization: schema.org JSON-LD (Person, 56 VisualArtwork, 2 CreativeWork, breadcrumbs), image sitemap, Open Graph/Twitter cards, canonical URLs, AI-crawler rules in robots.txt, llms.txt and llms-full.txt.
- Hardened the **Vercel** deployment: hash-based **Content-Security-Policy**, full security headers, 308 redirects, content-hashed long-term caching.
- Asset pipeline in **Python (Pillow)**: responsive WebP ~70% smaller, auto-rotation and cropping of 18 film scans, procedurally generated watercolor masks.
- Every release verified by a **50-check Puppeteer end-to-end suite** across **7 screen sizes** (320–1920 px).

Stack: React 18 · JavaScript (ES2020) · Node.js · esbuild · HTML5 · CSS3 · View Transitions API · Web Animations API · Python (Pillow) · Puppeteer · Lighthouse · Vercel

Phasora | Early stage (2 concept pages live) Mar 2026 – Present
Solo designer & developer
phasoraai.com

- A **physics-informed AI tutor** built on the **Anthropic API**. A custom physics-grounded system prompt enforces first-principles reasoning and returns structured math rendered with **KaTeX**.
- The tutor gives **hints instead of answers**.
- Features built so far: **interactive simulations** (spring-mass, projectile motion), **LaTeX formula cards** and an **AI problem generator**, on one canvas.
- Original pre-rendered physics animations made with **Manim** (Python); more are being added.
- Curriculum planned to cover all **11 units and 88 concepts** of General Physics I.

Stack: React · Vite · KaTeX · Anthropic API · Manim (Python) · Vercel

Hollow Alibi Oct 2026
Creator · AI-assisted build with OpenAI Codex
hollow-alibi.smoky-harp-3438.chatgpt.site · github.com/rmaytr/Hollow-Alibi

- A **2–10 player** game in **TypeScript, React 19 and Cloudflare Workers**, built around a server-authoritative state machine (**5 phases, 5 rounds**) that validates every action and computes all scores on the server.
- Hidden information stays private: each player gets a separate view built from server-only state, so opponents' cases, decisions and server randomness never reach a client before the reveal. Bots play from the same restricted view.
- No lost or double-counted moves during simultaneous play: **optimistic concurrency control** (version-checked writes to **Cloudflare D1/SQLite**), idempotent action receipts and client-side rejection of stale snapshots.
- Opt-in **WebRTC** group voice with authenticated signaling, reconnect-safe session epochs and **TURN** relay fallback. An automated **10-browser** test confirmed audio on all **90** directed paths, and a forced relay-only test confirmed audio in both directions.
- Account-free play secured with **SHA-256**-hashed session tokens in HTTP-only cookies, same-origin checks, request-size caps and per-session and per-IP rate limits. TURN credentials live in runtime secrets, never in client code.
- **3** bot difficulty levels, all rule-based; the Expert bot weighs public evidence, each opponent's honesty history and expected value.
- Redesigned the core loop after playtest feedback, replacing open-ended evidence reading with a claim-and-challenge system that gives every player a statement to defend and one named opponent to judge.
- **54** engine and storage tests, including all **8** payoff combinations, and **Playwright** end-to-end matches down to a 320 px screen. Fixed issues found in testing, including a timeout exploit that let players deny opponents' points.
- Accessible dialogs, visible focus, reduced-motion support and mobile-first judgment screens.

Stack: TypeScript · React 19 · Cloudflare Workers · Cloudflare D1 (SQLite) · Drizzle ORM · WebRTC (STUN/TURN) · Next.js 16 (Vercel build) · Playwright

Raincheck Raccoons | Entry, Handshake × OpenAI Multiplayer Game Challenge Oct 2026
Creator · AI-assisted build with OpenAI Codex
raincheck-raccoons.vercel.app · github.com/rmaytr/Raincheck-Raccoons

- A **2–10 player** game in **TypeScript, React 19 and Cloudflare Workers** with server-side rules, deadlines, scoring and host handoff. The frontend runs as **Next.js on Vercel** and talks to the Cloudflare game backend through signed server-to-server requests.
- Simultaneous moves stay consistent with revision-checked **Cloudflare D1** writes and bounded retries, plus round and match IDs on every choice. Tests with **9 concurrent joins** and **10 concurrent choices** resolved each round once with no duplicate scores.
- Account-free play secured with random session tokens stored as **SHA-256** hashes in HttpOnly cookies, **Zod** payload validation, a 4 KB body cap, same-origin checks and rate limits. Authentication moved ahead of quota writes after review found forged cookies could create database rows.
- Fixed a race where a late polling response could restore a room the player had just left, using a request-generation guard, and confirmed the fix with a browser test that held a real response until after leaving.
- Original **SVG** art (city scenes, **8** avatars, role icons) and browser-synthesized sound with the **Web Audio API**; responsive layouts checked at **320, 390, 768 and 1440 px**.
- Tuned the scoring rules with **18,000** simulated matches and a **5,280-case** role comparison, which led to raising one role's reward so it stayed a real strategic choice.
- Directed a multi-agent **OpenAI Codex** workflow from **100** game concepts to a **12**-concept shortlist, then defined API contracts and file ownership for backend, art/audio and QA agents, with independent review.

- **27** engine tests (**400** seeded scoring setups), **52** API and lifecycle checks, and **31 Playwright** checks run against the public deployment.

Stack: TypeScript · React 19 · Next.js 16 · Cloudflare Workers · Cloudflare D1 · Drizzle ORM · Zod · Tailwind CSS · Web Audio API · Playwright · Vercel

LeetCode Problems Repository 2026
Author
github.com/rmaytr/leetcode

- An AI-driven Python pipeline that documents and unit-tests solutions to 100 LeetCode problems (graphs, trees, 2D dynamic programming).

Stack: Python · Algorithms & data structures · Unit testing

Game Development, Unity & Unreal Engine 5 | In progress Summer 2026 – Present
Independent

- Prototyping games in Unity and Unreal Engine 5, gameplay systems, level design, real-time rendering.

Stack: Unity · Unreal Engine 5

UE5 Third-Person Adventure Game | 2nd of 30, ASU Game Development Camp 2024
Developer

- Built in Unreal Engine 5 with C++ and Blueprints during a 2-week camp; placed 2nd of 30 at Arizona State University's Game Development Camp.

Stack: Unreal Engine 5 · C++ · Blueprints

Python Chess Engine 2023
Author

- A full chess game in Python using OOP for game logic, move validation and rules.

Stack: Python · OOP

Game Automation Script 2023
Author

- A Python script that automates gameplay in Bloons TD6 over long sessions.

Stack: Python · Automation

LEADERSHIP & INVOLVEMENT

Events Chair, ACM at UAB Aug 2026 – Present (active member since Oct 2025)

- ACM at UAB is the local chapter of the world's largest computing society and the **largest CS organization on campus**.
- Plan and organize the year's events: **LeetCode nights**, **alumni panels**, **internship recruiting events** and **networking mixers**.
- Handle **outreach to companies** for event collaborations.
- Developing **joint programming with The AI Collective**.

Website Coordinator, Undergraduate Student Government Association (USGA) | USGA, Division of Communications Aug 2026 – Present

- In charge of the **official USGA website**, the main digital channel between student government and the student body.
- Build and maintain the site in **Joomla 5** with custom HTML, CSS and JavaScript.
- Planning and leading a **major redesign** of its structure, visual design and usability.

UAB Ambassador | Office of Student Engagement Apr 2026 – Present

- Selected as **1 of 16 new Ambassadors from 140+ applicants** for UAB Ambassadors (founded 1978), the university's official student hosts.
- Represent UAB at **high-profile university events** (Commencement, faculty and staff gatherings, events at the UAB President's Mansion) alongside senior administration, deans and guests.
- Recipient of a **$1,000** annual award.

International Mentor, International Scholar & Student Services (ISSS) Apr 2026 – Present

- Selected for a **competitive mentorship program** supporting UAB's international students from arrival through graduation.
- Co-lead **four 3-day New International Student Orientations** as part of a mentor team; greet students as they arrive in the U.S.
- Help new students with **official documents and administrative processes**.
- Meet weekly with **assigned mentees** and track their progress; hold office hours at the ISSS office.
- Help organize **large campus events** and create content for the program's social media.

- Recipient of a **$400 annual scholarship**.

Founder & President, The AI Collective at UAB (AIC) Feb 2026 – Present
AIC on CampusLabs

- Founded **UAB's first interdisciplinary AI organization**, advised by faculty in the Department of Computer Science; wrote the constitution and appointed the officers.
- Built a **4-officer executive board** and a **10-member founding team**.
- Designed **5 program tracks**: prompt engineering, productivity, coding and research literacy, AI ethics roundtables, industry briefings.
- First event this semester (Fall 2026); building partnerships with **ACM, USGA, Women in Technology**, faculty and alumni.

Freshman Forum Member, USGA · Member of the Year Sep 2025 – May 2026

- Selected as **1 of 31** from **100+ applicants** to represent the freshman class.
- **Conceived and founded the Micro-Internships initiative** with the **UAB Career Center**, handling the networking, employer outreach and program planning.
- Secured **$2,000 in scholarship funding** ($200 per intern) from on-campus departments and placed **10 students** in project-based micro-internships across **7 UAB departments** (Spring 2026).
- **Adopted long-term as Project One**, a cohort-based USGA and Career Center program with dedicated Career Center staff, its Fall 2026 cohort placed **20 students** at **18 departments**, including 3 off-campus.
- Remain a point of contact: helped match this year's students to sites and will speak at the closing ceremony.
- Led a **6-student team** with the Career Center Director to run a campus-wide **Career Readiness Week** (resume workshops, job fairs, headshots, professional development sessions).
- The **only member** in direct mentorship with the **USGA President**.
- Named inaugural **Member of the Month** (Oct 2025) and **Freshman Forum Member of the Year**.

Also: Women in Technology (WIT), member · UAB Tennis Club, member (2025 – Present)

EARLIER EXPERIENCE

Upper School Student Government Treasurer, Rancho Solano Aug 2024 – May 2025

- Managed **student store** financial operations and reconciled daily revenues exceeding **$300**.
- Organized major campus events (**Homecoming, Prom**) and fundraisers with the student council.

Journalist, School Newspaper "Pegasus", Rancho Solano | Scottsdale, AZ Aug 2024 – May 2025

- Authored **bi-monthly 500-word articles**, interviewing faculty and students to document campus life and events.
- Publicized new editions during convocations and integrated **multimedia content** to maximize readership engagement.

Varsity Tennis, 4th player, Rancho Solano Jan 2024 – May 2025

- **20+ matches** over 2 seasons; **"Most Coachable, Selfless, and Improved Player."**

Organizational Assistant & Actor, Arizona Russian Theater | Scottsdale, AZ Aug 2023 – Aug 2025

- Coordinated production logistics for **3 plays** and supported fundraising initiatives that secured over **$10,000** (total) for the nonprofit.
- Managed backstage operations and performed for audiences of **300+**.

School eSports President & Captain, Rancho Solano Aug 2023 – May 2025

- Managed operations for **two 5-person teams** competing in **HSEL and Vanta.gg** high school tournaments, coordinating practices and roster logistics.
- Analyzed gameplay recordings to formulate **data-driven strategies**.

Volunteer, Puerto Rico School Community Service Trip Aug 2023 – Aug 2025

- Restored local infrastructure by clearing forest debris and renovating a community basketball court to aid **disaster recovery**.
- Prepared and distributed **food supplies** to residents in need.

Boys volleyball, School #1354 | Moscow 2022

- **2nd of 8 teams** in an in-school tournament.

Peer Tutor & Academic Mentor, School #1354, Moscow 2021 – 2023

- Mentored **8 peers** in Informatics & Physics through over **100 coding problems** and **8 physics labs**.
- Prepared students for the **OGE** (main state exam), helping them achieve **"A" grades** in Computer Science & Physics.

Upper School Class President, School #1354, Moscow 2020 – 2023

- Represented a class of **38 students**, mediating academic and social concerns.

- Coordinated logistics for **7 educational field trips** and spirit events.

AWARDS & HONORS

At UAB

- **1st place**, **EcoCAR track**, HackBama_01, University of Alabama (Oct 2026)
- **Co-author**, AVS 72nd International Symposium abstract, accepted for an oral presentation (2026)
- **UAB Ambassador**, **1 of 16** new Ambassadors from 140+ applicants (2026)
- **Freshman Forum Member of the Year**, UAB USGA (2026)
- **Presidential Honors List**, UAB (2026)
- **Early admission**, UAB Accelerated Bachelor's-to-Master's Program in AI (2026)
- Inaugural **Freshman Forum Member of the Month**, UAB USGA (Oct 2025)

High school

- **Mu Alpha Theta** (National Math Honors Society), Student Officer; **highest AMC 12 score** at school (2023–2025)
- **2nd of 30**, **Game Development Camp**, Arizona State University (2024)
- **Prize Winner**, MISIS University Olympiad in Technical & Information Technologies (2023)
- **Prize Winner**, All-Russian Olympiad in Technology & Languages (2021–2023)
- **Award Winner**, Finskills Financial Competencies Championship (2022)

Certificates

- Anthropic: **Claude 101 · Claude Code 101 · Claude Platform 101**
- **Certificate in Python Programming**, Yandex Lyceum, Moscow (2023)
- **Ira A. Fulton Engineering Camp**, Arizona State University (2023)

SKILLS

Languages: Python · C · Java · JavaScript · TypeScript · HTML · CSS · Bash · Python asyncio

Web: React · Next.js · Astro · Node.js · Vite · Tailwind CSS · esbuild · Bootstrap · Joomla 5 · Vercel · Git · GitHub · GitHub Actions · Static pre-rendering · Structured data · hreflang · Sitemaps · WCAG · Three.js / WebGL · WebSockets · WebRTC · Cloudflare Workers · Supabase · Zustand · Zod · Drizzle ORM · PWAs · WebCrypto · SQLite · Docker · Google Cloud

Testing & performance: Playwright · Puppeteer · pytest · axe-core · Lighthouse · Chrome DevTools · Vitest · Hypothesis · mypy --strict

Scientific computing & ML: NumPy · Basilisk · Linux · HPC (Cheaha) · GP Bayesian optimization · Vehicle dynamics · CARLA · SageMath · Manim · LaTeX · KaTeX · Data visualization

AI tools: Anthropic API · Claude Code · Codex CLI · Skills & plugins · Parallel agents

Game development: Unity · Unreal Engine 5 · C++ · Blueprints

Research: IRB training · Data analysis · Research design · Excel

Spoken languages: Russian (native) · English (full professional) · Mandarin (beginner)