

# LUCAS McCLEAN

[lucas@mcclean.dev](mailto:lucas@mcclean.dev) | [linkedin.com/in/lucasmcclean](https://linkedin.com/in/lucasmcclean) | [github.com/lucasmcclean](https://github.com/lucasmcclean) | [mcclean.dev](https://mcclean.dev)

## EDUCATION

University of Central Florida Orlando, FL  
*Bachelor of Science in Computer Science* | GPA: 4.0 | *Burnett Honors College* Aug. 2024 – May 2027

## EXPERIENCE

**Software Developer** | *Python, Celery, Vue, Docker* Mar 2025 – Present  
*UCF Center for Distributed Learning - Techrangers Team* Orlando, FL

- Minimized tail latency for user-facing operations up to 70% by distributing large tasks to a separate queue
- Streamlined new-hire onboarding by clarifying failures through partial success states and refining error messages
- Improved enrollment system by enabling self-enrollment and integrating Canvas API for fallback user searches

**Sound & Lighting Technician** Jan 2022 – Jun 2023  
*South Tampa Fellowship* Tampa, FL

- Led a team of 3 in preparing and delivering sound, lighting, and visual content for audiences of up to 250
- Produced 5–6 weekly lighting scenes across multiple stages, balancing pre-programmed and live control
- Improved aesthetics and technical reliability by rewiring and staging new lighting and sound equipment

## PROJECTS

**LimitL.link** | *Go, PostgreSQL, Docker* Mar 2024 – Present

- Engineered a reliable link shortener with graceful recovery from runtime failures using channel signaling
- Architected a modular service layer to standardize database access and decouple business logic from storage
- Secured sensitive user data by generating high-entropy admin access tokens and using bcrypt to hash passwords

**Terminal Task** | *Go, Cobra* May 2024 – Jun 2024

- Ensured cross-platform compatibility by conditionally modifying behavior using Go's standard library
- Designed a modular API layer with four buffer and FS interfaces to maximize binary portability
- Implemented live terminal feedback by rendering output before screen refresh and isolating input logic per task
- Synchronized in-memory task order with render buffer to support consistent real-time updates
- Persisted independent task lists and configurations via direct integration with file system APIs

**Quick Note** | *Svelte, PostgreSQL, Docker* Mar 2024 – Apr 2024

- Implemented a full-featured CRUD API with PostgreSQL backend and containerized local deployment
- Leveraged Svelte's component system to unify styling and streamline frontend reuse
- Synchronized UI state with backend by confirming API calls prior to DOM updates

**Crown & Anchor** | *JavaScript, HTML, CSS* Feb 2024 – Mar 2024

- Designed multi-page browser game preserving session state across reloads and internal navigation
- Utilized the localStorage API to persist game state and metadata across browsing sessions
- Modularized reusable UI components to streamline development and maintain behavioral consistency

**Doomsday Catastrophe** | *Godot, GDScript* Nov 2023 – Nov 2023

- Directed a 5-person team to complete a full game aligned with contest specifications in under 7 days
- Structured 10+ game objects using OOP principles to support maintainable logic and collaborative workflows
- Developed a fast level generator producing 15 rooms every 3 levels in under one second
- Built state persistence using JSON-based save/load system with direct file system integration

## TECHNICAL SKILLS

**Advanced:** Go, HTML/CSS, Git, Docker, Neovim, Tmux, Linux (Arch, Gentoo, Fedora)

**Intermediate:** Rust, Python, JavaScript, C, Godot, Cargo, PostgreSQL

**Familiar:** Lua, Bash, SQL, Celery, Svelte, Vue, MongoDB, Redis