# ABHIJEET S PACHPUTE

abhijeetsp21@gmail.com | (801)949-7940 | linkedin.com/in/abhijeet-pachpute/ | abhijeetpachpute.com

## Summary

Adaptable computer science graduate and software engineer with experience building scalable systems, optimizing workflows, automating decision-making. Proven ability to rapidly master new technologies, solve complex problems, deliver measurable product improvements. Strong communicator with a track record of translating technical solutions for diverse stakeholders.

## Education

**University of Utah** *August 2024 - May 2026*
*Master of Science in Computer Science | GPA: 3.7/4.0*
***Relevant Coursework:*** Graduate Algorithms (CS 6150), Deep Learning (CS 6353), Security & Privacy (CS 6495)

**University of Pune** *July 2019 - May 2023*
*Bachelor of Engineering in Computer Engineering | CGPA: 8.7/10*
***Relevant Coursework:*** Object-Oriented Programming, Database Management, Data Structures & Algorithms, ML

## Skills

**Programming Languages**: Java, Python, C++, C, Kotlin, JavaScript (ES6+), TypeScript

**Backend & Databases**: Spring Boot, RESTful APIs, caching, concurrency, Node.js, MySQL, MongoDB

**Cloud & Infrastructure**: AWS (EC2, S3, Lambda), Docker, Kubernetes, CI/CD, Git, GitHub

**Frontend Technologies**: React.js, Next.js, Tailwind, HTML, CSS

## Experience

**Information Technology Intern** *June 2025 - Present*
*University of Utah | VP for Research* *Salt Lake City, USA*

- Administered research infrastructure across 4 departments that powered a **$650M+** annual enterprise and deployed security solutions: **Microsoft Intune, Tanium & BeyondTrust** that reduced risk by **60%** and secured compliance with federal research data standards
- Formalized and co-authored **SOPs** for international loaner laptops and onboarded **70+ devices** with full-disk encryption, bringing deployments into compliance with university policies

**AI Software Engineer Intern** *May 2025 - August 2025*
*AVI Human Services* *Remote, USA*

- Led development & deployment of real-time AI analytics using React, **Node.js, TypeScript**, Gemini API, **Docker**, **AWS**
- Enabled intelligent querying and automated alerts for **50,000+** student records across 10 U.S. state agencies on **production grade, fault-tolerant backend** services
- Designed and optimized **RAG**-based **prompt orchestration** and backend services using **vector embeddings** and **Kubernetes**, improving content relevance by **40%** and enabling scalable, automated curriculum generation

**Software Developer Intern** *July 2023 - February 2024*
*eWarranty Solutions* *Pune, India*

- Developed a large-scale **QR code** based warranty **verification** system using **Java, Kotlin**, Spring Boot, and **RESTful APIs**, streamlining workflows for over **45,000** products and cutting manual errors by 30%
- Engineered **real-time analytics** dashboard with **Spring Boot**, JavaScript, **MySQL**, and integrated CompletableFuture API for asynchronous data retrieval, enabling manufacturers to make latency sensitive, data-driven warranty decisions
- Enhanced backend performance by optimizing database queries, **API workflows**, and system architecture with MySQL, **Kotlin** Coroutines, reducing response times by 40% and improving system stability under concurrent load

## Projects

**Dinodash** | *C++, Raylib, OpenGL, Clang/GCC, Cross-platform* | Website

- Developed 2D survival game in **C++** using **Raylib** & **OpenGL** with real-time physics, collision detection, & dynamic object spawning while maintaining stable **60 FPS** performance through careful memory & resource management
- Implemented advanced graphics systems including multi-layer **parallax scrolling**, procedural day-night cycles, & optimized resource allocation for **25+** game assets with proper resource cleanup

**Custom Unix Shell** | *C, System Calls, Process Management, File I/O* | GitHub

- Implemented fully functional Unix-like shell in **C** supporting command execution, piping, I/O redirections (`>, >>, <`), background processes (`&`), and logical operators (`&&`, `||`) using **fork(), exec(), and wait()** system calls
- Designed asynchronous background process execution with **PID tracking** & file descriptor management, achieving stable performance across **500+** test commands with proper cleanup & edge case handling under concurrent process execution