Karan Jadhav

Pune, Maharashtra | M: +91-8669790435 | E: karanjadhav0027@gmail.com | GitHub | LinkedIn

Professional Summary

A detail-oriented and motivated recent Master of Computer Applications (MCA) graduate with a **9.36 CGPA** and a strong foundation in backend development using Java, Spring Boot, and Python. Proficient in data structures, algorithms, object-oriented programming, and the full software development lifecycle. Eager to apply problem-solving and technical skills to develop robust and scalable software solutions.

Education

- **Master of Computer Applications (MCA)** | Dr. D. Y. Patil College of Management & Research, Pune | *Savitribai Phule Pune University* | **9.36 CGPA** | 2023 – 2025
- **Bachelor of Computer Science (BCS)** | MSG College, Malegaon | *Savitribai Phule Pune University* | **6.5 CGPA** | 2020 – 2023
- **HSC (Science)** | MSG College, Malegaon | *Maharashtra State Board* | **69.83%** | 2020

Core Competencies

- **Languages:** Java, Python, C++, C, SQL
- **Backend & Frameworks:** Spring Boot, Spring MVC, REST APIs
- **Databases:** MySQL, MongoDB
- **Web Technologies:** HTML, CSS, JavaScript, ReactJS
- **Data Science & ML:** NumPy, Pandas, Matplotlib, Scikit-learn (Regression, Classification)
- **Developer Tools & Methodologies:** Git, GitHub, Maven, Gradle, IntelliJ IDEA, VS Code, Agile, OOP, Design Patterns
- **Operating Systems:** Windows, Linux (Ubuntu)

Technical Projects

Duplicate File Remover with Automated Email Logs | Technologies: Python

- Engineered a robust Python script to identify and delete duplicate files within a directory by comparing MD5 checksums, ensuring data integrity.
- Implemented automated log generation and scheduled email delivery of results using SMTP, providing system status without manual intervention.
- Developed a command-line interface (CLI) for user-friendly input of directory paths, scheduling intervals, and recipient email addresses.

Batch Entry Microservice API | Technologies: Java, Spring Boot, MongoDB, REST API

- Engineered a scalable microservice for managing batch records using Spring Boot and MongoDB with complete CRUD support.
- Designed RESTful endpoints following industry-standard API conventions, enabling seamless integration with frontend or cloud systems.
- Implemented controller-service-repository architecture and incorporated health-check endpoints for uptime monitoring.
- Built cloud-ready, containerizable codebase (Docker-compatible) with Git-based version control, supporting CI/CD pipelines.

Customised Virtual File System (VFS) | Technologies: C, Data Structures

- Designed and implemented a VFS in C, replicating core functionalities of the Linux file system, including custom shell commands and system call implementations.
- Engineered all essential file system data structures from scratch, such as the In-core Inode Table (II-Table), File Table, UAREA, and User File Descriptor Table.
- Enabled cross-platform execution, allowing Linux file system operations to run on other operating systems through a customized shell and database.

Secure File Packer & Unpacker | Technologies: Java, Swing, Cryptography

- Developed a desktop application using Java Swing to pack multiple files into a single archive and unpack them, complete with metadata management.
- Integrated symmetric encryption/decryption algorithms to secure the contents of packed files, ensuring data confidentiality.
- Designed an intuitive Graphical User Interface (GUI) for seamless user interaction during packing and unpacking operations.

Peer-to-Peer Chat Messenger | Technologies: Java, Socket Programming

- Built a platform-independent, text-based chat application for peer-to-peer communication using Java's socket programming and multithreading.
- Implemented a logging mechanism to automatically record chat history with timestamps for audit and review purposes.

Generic Standard Template Library (STL) | Technologies: C++, OOP

- Created a comprehensive library of generic data structures (Linear and Non-Linear) and algorithms in C++.
- Applied template programming to ensure the library components were reusable for any data type, enhancing modularity and development efficiency.