JAVED ANSARI

BACKEND DEVLEOPER

j.ansari6255@gmail.com | +977-9818840608 | www.javedansari.com.np | https://github.com/zyavhed

SUMMARY

Specializes in Python-based backend development, Odoo customization, database operations, role-based access control, data migration, cron-based automation, and asynchronous processing. Experienced in translating business requirements into practical technical solutions while collaborating directly with clients and internal teams. Seeking to contribute strong problem-solving, backend engineering, and ERP development expertise to a forward-thinking technology organization.

SKILLS

Languages: Python, C, C++, JavaScript, SQL
Frameworks & Platforms: Odoo, Django, Flask
Backend Development: REST APIs, Server-Side Development, Authentication, Role-Based Access Control, Background Jobs, Cron Jobs, Batch Processing
Odoo Development: Odoo ORM, Module Customization, QWeb Reports, Automated Actions, Scheduled Actions, Data Migration, Workflow Automation
Databases: PostgreSQL, MySQL, Database Design, SQL Query Optimization
Cloud & DevOps: AWS Solutions Architect – Associate (SAA-C03), Linux, Git, GitHub
Data & AI: NLP, N-gram Tokenization, Similarity Algorithms, Data Processing
Computer Science: OOP, Data Structures, Algorithms, DSA, Software Engineering

WORK EXPERIENCE

Odoo Developer | Smarten Technologies **Nov 2025 - Present**

- Customized 5+ Odoo modules across Sales, CRM, Inventory, POS, and eLearning to automate business workflows.
- Built server-side data migration pipelines for large client datasets, resolving data inconsistencies and dependencies.
- Automated duplicate detection, record merging, pricing rules, and validations using cron jobs and automated actions.
- Improved heavy imports and reporting with batch processing and asynchronous job queues.
- Worked directly with clients on requirements and applied AWS architecture principles for scalable and reliable solutions.

Trainee | Bajra Technologies **Jun 2023 - Sept 2025**

- Built Python-based projects applying OOP, data structures, functional programming, debugging, and testing principles.
- Designed and optimized SQL queries and database schemas using PostgreSQL/MySQL, improving understanding of relational data modeling and backend development.
- Worked with Linux and Git/GitHub-based collaborative workflows, contributing to [X+] development tasks/projects.
- Gained foundational experience in Odoo development and ERP customization, progressing into professional Odoo development after the traineeship.

EDUCATION

Pulchowk Campus of Engineering (Tribhuvan University) **2021 - 2025**
Bachelors in Computer Engineering with overall score 70.24% (First Division)

Nepal Mega College **2019-2020**
Completed +2 in Science with an overall GPA of 3.51.

Lyceum Paradise Academy **2018**
Completed Secondary Education Examination (SEE) with Overall 3.65 GPA

PROJECTS

NLP-Based Database Query System — 2024
Python | SQL | NLP | LCS | N-gram Tokenization | Keyword Extraction

- Developed an NLP-driven query system that converts natural language questions into executable SQL queries, allowing users to interact with relational databases without manually writing SQL.
- Designed a custom text preprocessing and tokenization pipeline that normalizes user input and generates unigram and n-gram representations for downstream query analysis.
- Implemented database table identification using Longest Common Subsequence (LCS) and neighboring similarity scoring to determine the table most relevant to the user's query.
- Developed column-matching logic using similarity thresholds to map natural-language terms to database fields and automatically construct SQL SELECT and WHERE clauses.
- Implemented support for multi-condition queries using AND / OR operators and six comparison operators, including equality, inequality, greater-than, less-than, and their inclusive variants.
- Added automatic data-type detection to distinguish between numeric and string values when generating SQL conditions.
- Implemented ORDER BY query generation by identifying natural-language sorting keywords such as ascending and descending and mapping them to the appropriate database columns.
- Designed the system as a modular pipeline separating tokenization, table selection, column identification, condition extraction, and SQL generation, improving maintainability and extensibility.
- Focused on handling variations in user input through approximate string matching and similarity-based decision making rather than relying solely on exact keyword matches.

PDF Plagiarism Detection System — 2024
Python | PyMuPDF | PyPDF2 | NLP | Cosine Similarity | Manhattan Distance | Euclidean Distance

- Developed a document-analysis system capable of detecting potential plagiarism across multiple PDF documents using automated text extraction and similarity analysis.
- Built a PDF processing pipeline using PyPDF2 and PyMuPDF to extract, process, compare, and annotate document content.
- Implemented unigram, bigram, and trigram tokenization and combined similarity results across different n-gram levels to improve comparison robustness.
- Implemented Cosine Similarity from scratch, constructing word-frequency vectors and calculating vector dot products and magnitudes to derive document similarity scores.
- Added Manhattan Distance and Euclidean Distance as alternative similarity algorithms, allowing users to evaluate documents using different mathematical approaches.
- Converted similarity measurements into a 0–100 percentage-based score and implemented a configurable threshold for determining when documents should be flagged as potentially plagiarized.
- Developed a transitive clustering mechanism that groups documents with overlapping plagiarism relationships, ensuring indirectly connected documents are represented within the same plagiarism group.
- Implemented extraction of common 5-gram phrases between documents to identify the specific textual patterns contributing to similarity.
- Used PyMuPDF annotations to highlight matching phrases directly within the original PDF documents, generating annotated copies for easier review by instructors.
- Structured the system to separate document extraction, tokenization, similarity calculation, clustering, and PDF annotation, making individual components easier to modify and test.

Baagchal Game using SFML — 2023
C++ | SFML | Object-Oriented Programming | Game Logic

- Developed a desktop implementation of Baagchal, the traditional Nepali strategy board game, using C++ and object-oriented programming principles.
- Designed the complete game engine and rule system, including player turns, piece movement, valid-position checking, captures, and win-condition detection.
- Implemented game-state management to maintain and update the board dynamically after every player action.
- Developed move validation logic to prevent invalid movements and enforce the rules governing both tigers and goats.
- Built an interactive graphical interface using SFML, integrating user input, board rendering, piece movement, and real-time game-state updates.
- Applied object-oriented design to separate game entities, board state, movement logic, and user interaction, resulting in a modular and maintainable codebase.
- Implemented the game loop and event-handling system required for real-time desktop interaction.
- Focused on maintaining clear separation between the underlying game logic and graphical presentation, allowing the core game mechanics to operate independently of the UI.