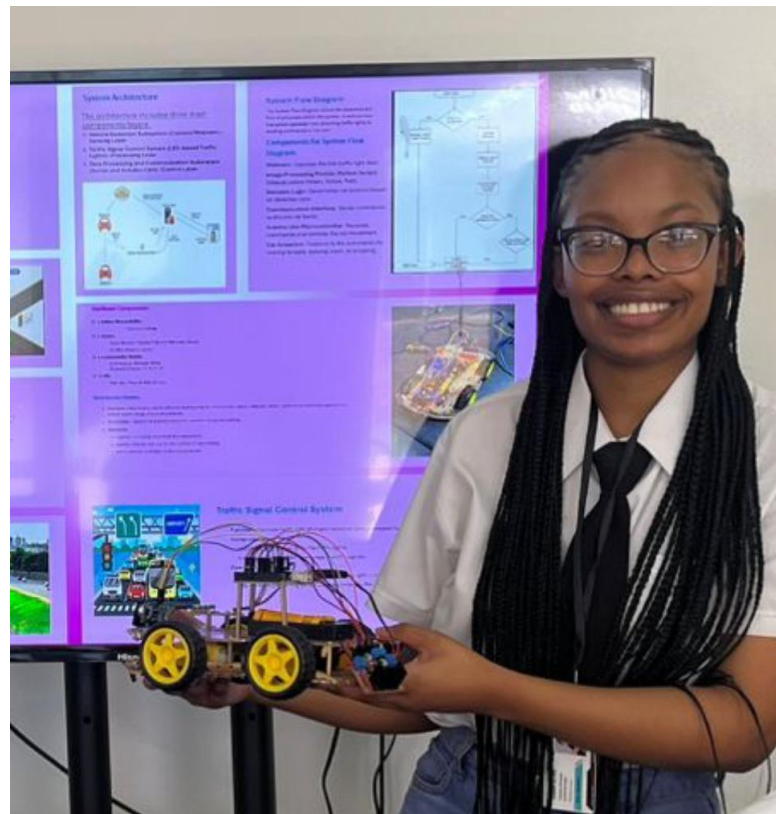


Project Portfolio – Yolanda Msingale

Links:

- **GitHub:** [GitHub Link](#)
- **Personal Portfolio Website:** [Website URL](#)
- **LinkedIn:** [LinkedIn URL](#)



Overview

This document presents a curated selection of academic and personal software projects that showcase my practical experience with technologies such as **C#**, **.NET**, **Visual Studio**, **MS SQL Server**, and **Git**. Each project reflects my ability to apply core programming concepts, solve real-world problems, and build functional applications across desktop, web, and database environments.

All projects included here were developed either during my studies or through self-directed learning, and they collectively demonstrate my readiness for a professional IT environment. The source code for each project is available via the links.

Projects

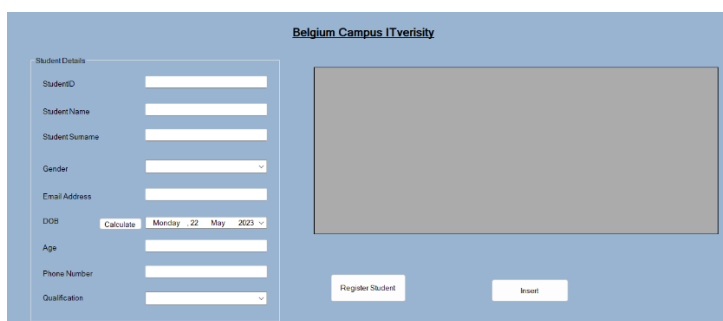
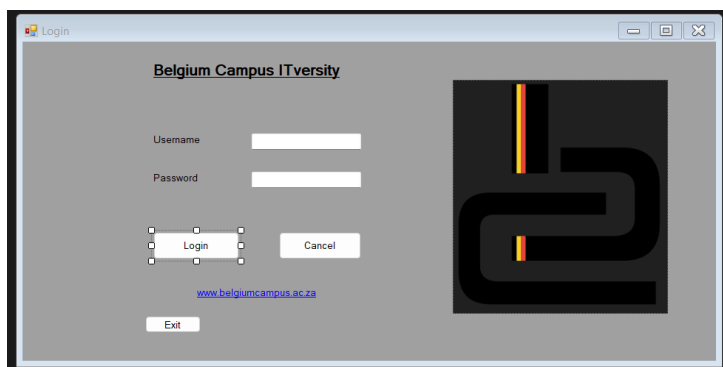
Student Registration System

Tech Stack: C# (Windows Forms), Visual Studio

Key Concepts: Object-Oriented Programming (OOP) – Inheritance, Encapsulation, Abstraction

GitHub URL: [Project Link](#)

This project focused on building a Windows Forms desktop application designed to manage and process student information efficiently using C# and Visual Studio. The primary objective was to develop a functional, intuitive registration system that applies foundational Object-Oriented Programming (OOP) principles throughout.



Key Features:

- **Student Registration:** Allows users to create and store new student profiles.
- **Course Enrollment:** Enables students to be assigned to specific academic programs.
- **Record Management:** Provides administrators with tools to view, edit, or delete student data through a centralized interface.
- **User-Friendly Interface:** Implemented using Windows Forms, offering a clean, responsive experience tailored to administrative users.

OOP Principles Applied:

- **Encapsulation:** Critical data such as grades and personal information is safeguarded through private fields, with access managed via public getters and setters.
- **Inheritance:** Introduced a generalized Person class, from which Student and Administrator classes inherit, promoting code reuse and logical structure.
- **Abstraction:** Core functionalities such as registration logic and validation were abstracted into interfaces and base classes to enhance modularity and simplify future maintenance.

This system not only helped reinforce my understanding of C# and Windows desktop development, but also sharpened my architectural thinking in designing scalable and maintainable applications. It was a hands-on application of theoretical concepts learned in class, bridging the gap between coursework and real-world software development.

 Pizza Restaurant Ordering System

Tech Stack: Database Administration (SQL, ER Modelling)

GitHub URL: [Project Link](#)

This project involved designing and implementing a **relational database system** for a local pizza restaurant to streamline operations such as order tracking, customer management, menu item organization, and delivery scheduling.

Key Features:

- **Customer Management:** Stored customer information including contact details and order history.
- **Menu Items:** Created tables to manage dynamic menu items with pricing, descriptions, and availability status.

- **Order Processing:** Designed relationships between customers, menu items, and orders to enable accurate, real-time order tracking.
- **Delivery Tracking:** Included functionality to assign and track deliveries with timestamps and delivery personnel.

Database Design:

- **ER Diagrams:** Constructed detailed Entity-Relationship diagrams to visualize and define relationships between data entities.
- **Normalization:** Applied normalization techniques (up to 3NF) to eliminate redundancy and maintain data integrity.
- **SQL Queries:** Wrote efficient SQL scripts for table creation, data insertion, and essential operations like querying order history, generating sales reports, and tracking popular items.

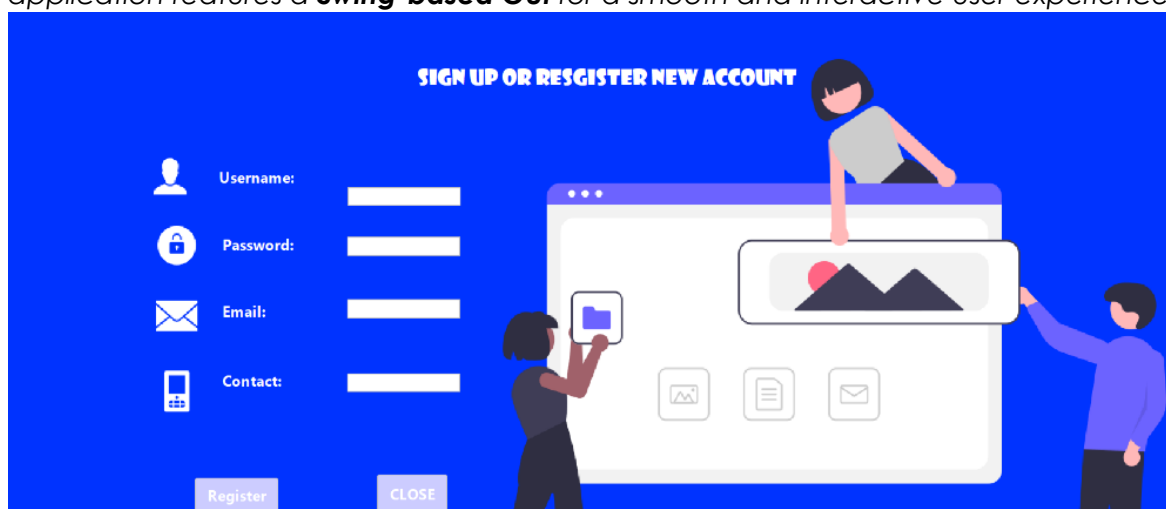
This project enhanced my understanding of **database design principles, data modelling, and SQL programming**. It also gave me practical experience in planning a database from the ground up to support a real-world business scenario.

 Java Project – Library Management System

Tech Stack: Java, Swing, JDBC, PostgreSQL, JSP, Servlets, Apache Tomcat, GlassFish

GitHub URL: [Project Link](#)

This project demonstrates the integration of **Core Java concepts** with both front-end and back-end technologies, showcasing a full-stack Java-based application. The application features a **Swing-based GUI** for a smooth and interactive user experience.



LOG IN

Username:

Password:



WELCOME

 **AUTHORS**

 **BOOK DETAILS**

 **BORROWERS**

 **LOG OUT**

AUTHORS DETAILS

Authors ID:

Name:

Surname:

Book Written:

Title 1	Title 2	Title 3	Title 4
---------	---------	---------	---------

Key Features:

- **User Interface:** Built a responsive desktop GUI using **Java Swing**, allowing users to perform tasks like data entry and record management easily.

- **Database Integration:** Connected to a **PostgreSQL** database using **JDBC** for executing secure and efficient **CRUD operations**.
- **Backend Development:** Utilized **Java Servlets** and **JSP (Java Server Pages)** to manage server-side logic and dynamic content generation.
- **Deployment:** Deployed the application on **Apache Tomcat** and **GlassFish** servers for testing and demonstration.

Development Focus:

- Applied **Object-Oriented Programming (OOP)** principles such as inheritance, abstraction, and polymorphism throughout the application structure.
- Implemented form validation, error handling, and data consistency checks to ensure a reliable user experience.
- Designed modular code with MVC-inspired separation between UI, logic, and data access layers.

This project deepened my skills in **Java development**, **GUI design**, **server-side programming**, and **database connectivity**, and provided hands-on experience in deploying real-world Java applications.



Tech Stack: Python, Tkinter, SQLite

GitHub URL: [Project Link](#)

This project involved building a **desktop-based management system** using **Python and Tkinter**, aimed at simplifying administrative tasks for managing both students and lecturers. The user interface was designed to be intuitive, allowing easy navigation between different modules.

Key Features:

- **Student Management:** Add, view, update, and delete student records including personal details, course enrolment, and performance tracking.
- **Lecturer Management:** Add, view, update, and delete Lecturer records including personal details just like in student management
- **GUI:** Created a multi-window interface using **Tkinter**, ensuring smooth interaction and a clear user experience.

Database Integration:

- Used **SQLite** as the backend to store and manage application data.
- Performed **CRUD operations** via Python's built-in `sqlite3` module to ensure lightweight and reliable data handling.

- Designed a normalized database schema to avoid redundancy and improve query performance.

This project enhanced my practical knowledge of **Python programming, desktop GUI development, and relational database integration**. It also provided hands-on experience in building end-to-end applications that mimic real-world administrative systems.

WordPress Blog

Tech Stack: WordPress, PHP, MySQL, HTML, CSS

WordPress Blog: [WordPress Link](#)

GitHub URL: [Project Link](#)

This group project involved designing and deploying a blog site called "The Culers", built with the WordPress CMS. The blog serves as a team portfolio for a Belgium-based group of five members, highlighting individual profiles and team leadership, while fostering collaboration and personal expression.

Key Features:

- **Team Introduction Layout:** Structured the homepage to introduce all group members—Lihle, Yolanda, Chantel, Rorisang, and Tlhologelo—with roles and brief personal insights.
- **Custom Content Pages:** Each member has a dedicated section detailing strengths, weaknesses, favorite movies, and personal likes to build a personable, relatable blog presence.
- **Theme Customization:** Edited and styled a WordPress theme using PHP, HTML, and CSS to match the group's identity and branding.
- **Plugin Integration:** Added social media links (Instagram, Facebook, Twitter), contact tools, and SEO plugins (e.g., Yoast) to improve functionality and engagement.
- **Mobile-Responsive Design:** Ensured the blog is readable and navigable on both desktop and mobile devices for broader accessibility.
- **Search and Navigation Enhancements:** Customized menus and search functions to allow easy content discovery and improve overall user experience.

Database & Backend:

- **MySQL Integration:** Utilized WordPress's backend structure powered by MySQL to store posts, comments, media, and user data.
- **Dynamic Content Display:** Learned how PHP templates interact with MySQL to render user-generated and dynamically updated content throughout the site.

Project – Portfolio – Yolanda Msingale

This project deepened my skills in WordPress customization, content management, front-end styling, and basic PHP development. It also showcased my ability to collaborate within a team and contribute to the development of a visually consistent and technically functional group website.

Software Testing

Tech Stack: Java, SQL, Waterfall SDLC, JUnit, Manual Testing

GitHub URL: [Project Link](#)

This project involved collaborating with the Belgium Campus PRG282 Software Development team to conduct structured testing on a **Student Information Management System**. The goal was to ensure the system's stability, correctness, and alignment with the original business requirements throughout the software development lifecycle.

Key Features:

- **Test Planning:** Developed and followed detailed test plans and test cases based on functional specifications.
- **Manual Testing:** Performed hands-on testing of core features like student registration, data retrieval, and record updates to identify functional and UI defects.
- **JUnit Testing:** Used JUnit to implement unit tests for backend logic and data operations written in Java.
- **Bug Reporting:** Logged issues and feedback systematically to the development team for fixes and enhancements.
- **Waterfall SDLC Evaluation:** Tested the software at key stages of the Waterfall model, ensuring each phase met its deliverables before proceeding.

This project sharpened my attention to detail and deepened my understanding of how quality assurance fits into the software development pipeline — a critical skill in both frontend and full-stack development roles.

Web Programming Project - Issue/Bug Tracking System

Tech Stack: JavaScript, HTML, CSS, Node.js, Express, MongoDB

GitHub URL: [Project Link](#)

This project involved building a lightweight issue tracking web application, designed for managing tasks and bugs within a software development workflow. Developed as part

of an academic project, the application helps teams document, track, and resolve issues systematically, simulating real-world software project collaboration.

Key Features:

- **Issue (Ticket) Management:**
Create, view, update, and assign issues with detailed metadata, including summary, description, project, priority, assigned person, and resolution status.
- **Status Tracking:**
Each issue includes a progress indicator (Open, Resolved, Overdue), with built-in logic to track target vs. actual resolution dates.
- **Project and Team Setup:**
Allows for static creation of people (ID, name, email, profile picture) and projects (ID, name) to assign tasks appropriately.
- **Single Sign-On:**
Uses a basic admin login for secure access to the system's features.
- **Interactive UI:**
Clean and intuitive user interface built with HTML, CSS, and JavaScript. Bootstrap was used to enhance responsiveness and aesthetic quality.
- **RESTful API (Bonus Implementation):**
A Node.js and Express backend was optionally created to simulate a more scalable solution using real-time data persistence with MongoDB.
- **LocalStorage Version:**
In adherence to academic constraints, the core version of the app uses Web Storage API (localStorage) to store data such as tickets, people, and project information locally across sessions.
- **Markdown Documentation:**
An integrated Markdown page provides system documentation, setup instructions, and technical overview as required by the course deliverables.

This project was developed in response to a scoped academic brief, which emphasized essential issue tracking functionality. It serves as a foundational full-stack web app, laying the groundwork for future enhancements such as authentication, user roles, analytics, and deployment on cloud platforms.

 Database Functionality Project

Tech Stack: Microsoft Access (Tables, Queries, Forms, Reports, Macros), VBA (Visual Basic for Applications)

GitHub URL: [Project Link](#)

This project involved building a comprehensive **Integrated Student Management System** using Microsoft Access, serving as both the frontend and backend. The system supports various academic program types and learning models while ensuring accurate record-keeping and streamlined data handling.

Key Features:

- **Student Profile Management:** Record and manage personal information, course enrolment, subject selections, and payment details.
- **Course & Subject Tracking:** Supports multiple course types (certificate, diploma, degree, skills) and study models (full-time, part-time, distance).
- **Payment & Book Allocation:** Tracks payment history and allocates textbooks based on course requirements.
- **Custom Forms & Reports:** Created user-friendly forms for data entry and dynamic reports for administrative insights.
- **Automation & Validation:** Used VBA for input validation, custom business logic, and automating repetitive tasks (e.g., overdue payment alerts).

Relational Database Design:

- Built a normalized relational database with linked tables such as **Students, Courses, Subjects, Payments, and Books**.
- Ensured data integrity through defined relationships and enforced referential integrity rules.
- Designed queries to extract meaningful insights and reports for academic and financial monitoring.

This project enhanced my understanding of practical database systems and gave me hands-on experience in creating real-world administrative tools using Microsoft Access.

 Vehicle Infrastructure Integration

Tech Stack:

- **IoT Devices & Sensors** – For collecting real-time data (speed, signals, traffic density)
- **Edge Computing & Embedded Systems** – For local, low-latency data processing
- **V2X Communication** (DSRC/C-V2X) – Vehicle-to-Infrastructure data exchange
- **Cloud Platforms** – AWS, Azure (for large-scale data storage and analysis)
- **Machine Learning & AI** – For traffic prediction and anomaly detection
- **Programming Languages** – Python, C/C++, Java
- **GIS & Mapping Tools** – For route planning and traffic visualization

GitHub URL: [Project Link](#)

This advanced project focused on **integrating vehicle infrastructure** with smart systems to enhance road safety, traffic efficiency, and real-time decision-making through connected technology and AI.



Key Features:

- **Enhanced V2I Communication:** Implemented systems for seamless data exchange between vehicles and infrastructure (e.g., traffic lights, road sensors).
- **Real-Time Processing:** Deployed edge devices for processing critical data locally to reduce latency and support time-sensitive decision-making.
- **Traffic Flow Optimization:** Used AI models to predict congestion, adjust traffic signal timings, and prioritize emergency vehicles.
- **Cloud Integration:** Transmitted and stored traffic data on cloud platforms (AWS/Azure) for analysis and long-term planning.
- **Mapping & Routing:** Integrated GIS tools to visualize traffic patterns and suggest optimized routes.

What I Learned:

- Developed hands-on experience with **V2X communication protocols** and IoT integration.
- Learned how to combine **machine learning** with real-world sensor data for predictive analytics.
- Improved understanding of **edge vs. cloud processing**, and when to use each effectively.

- Gained exposure to **embedded systems, data security, and smart city technologies.**
- Applied **multi-language programming skills** (Python, Java, C++) across components.

This project deepened my knowledge of intelligent transport systems and showcased how emerging technologies can transform traditional infrastructure into smart, adaptive networks.

Why we should hire you?

The projects outlined in this document reflect both my **academic training** and my commitment to **continuous self-development in IT**. I actively engage in **personal projects** to expand my skills, stay updated with modern technologies, and apply knowledge in practical, real-world settings.

I am currently **developing websites and applications for individuals starting their own businesses**, helping them turn ideas into fully functioning digital platforms. These experiences have strengthened my ability to work on real-world use cases, meet client requirements, and deliver complete solutions — all while continuing to grow as a developer.

Skills Snapshot:

Skills:

- **Programming:** Python, Java
- **Web Development:** HTML, CSS, JavaScript
- **Database Management:** MySQL, PostgreSQL, MongoDB
- **Tools & Platforms:** VS Code, NetBeans, GitHub, Docker

Soft-Skills:

- Team Collaboration
- Time Management
- Adaptability
- Quick Learner
- Leadership

Extra-curricular Activities

- Participated in weekly coding challenges and hackathons
- Conducted peer tutoring sessions on Python and web development
- Contributed to multiple open-source projects via GitHub
- Engaged in smart city and AI-driven hackathons

Internship/IT-related position Objective

I am currently seeking an **IT internship or IT-related position** to complete the final module (AIT361) of my Information Technology qualification. I have successfully completed all academic coursework, and this opportunity will allow me to gain essential industry experience while contributing meaningfully to your organization.

With a strong technical foundation and a passion for learning, I bring not only practical development skills but also the adaptability and teamwork needed in a fast-paced professional environment. I'm ready to take on new challenges and deliver real value from day one.