Google Summer of Code 2025 Project Proposal

Personal Information

- **Name:** Aime Pacifique Abayisenga
- **LinkedIn:** LinkedIn Profile
- **Email:** paccy7002@gmail.com
- **GitHub Username:** PaccyC
- **Contact Number:** (+250) 796168499
- **Primary Language:** English
- **Institution:** Rwanda Coding Academy

About Me

I am Aime Pacifique Abayisenga, a dedicated software developer with a strong passion for building scalable and maintainable software solutions. With over three years of experience in software development, I have worked with various modern technologies, including JavaScript, TypeScript, React.js, Node.js, Express.js, and MySQL for backend development.

I am always eager to learn and contribute to open-source projects. My experience includes building real-time applications and scalable backend systems. I actively participate in coding challenges on platforms such as **LeetCode** and engage with developer communities to improve my skills.

Availability During the Coding Period

The only prior commitment I have is an important **exam from July 9th to July 18th**.
Outside of this period, I will be **fully dedicated** to the project, contributing **5-7 hours per day**.

Relevant Experience

Projects

Etix – Smart Bus Ticketing System

- **Objective:** Modernized Rwanda's transport sector by enabling passengers to book tickets and track bus arrivals in real time, reducing delays and improving efficiency.
- **Technologies:** React Native, Node.js, Jest, Docker
- **Role:** Backend Developer

Sarura – Smart Agriculture Solution

- **Objective:** Assisted farmers in maximizing yield using AI-powered early plant disease detection, automated pesticide spraying, and real-time farm condition monitoring.
- **Technologies:** React Native, Spring Boot, Django, JUnit, Mockito, Microservices Architecture
- **Role:** Backend Developer

Project Overview

Project Title: MyUni – A Unified University App for Greek Universities

Abstract

Many Greek universities use separate digital platforms to manage student information, university announcements, and essential services. While some universities have implemented solutions like **MyUoM (University of Macedonia) and UniWA (University of West Attica)**, these platforms lack:

- A **scalable** backend architecture
- A **modular framework** that other universities can easily adopt
- **Real-time updates** fetched from official university sources

This project aims to build a **scalable, modular**, and **customizable** platform that Greek universities can integrate into their systems. The project will include a **custom CMS**, a robust **backend architecture**, and a **student portal** for managing academic data, announcements, and university-related information.

Benefits to the Community

1. **Standardization of University Apps** – A unified and modular framework that all Greek universities can integrate with minimal effort.
2. **Real-time Updates** – Synchronization of university announcements, schedules, and official notifications directly from university sources.
3. **Scalable Backend** – A robust and scalable architecture capable of handling thousands of students.
4. **Customizable CMS** – Universities can easily manage and update their content without modifying code.
5. **Improved Student Experience** – A **personalized student dashboard** where students can access their academic records, announcements, and schedules.
6. **Multi-Domain Support** – The app will support multiple universities with independent customization per institution.

Alignment of project with my future goals

During my time in programming, I have worked on different projects through hackathons and other programs and I am eager to contributing in open source development which will give me the chance to work on applications that can be used by millions of users and I'll be able to

extend the borders to reach far away from my home country.Through the program I will be able to learn new skills including new architectures used in development of softwares,improve ability to write scalable codes and many other things from different people especially mentors.

As I have those goals, I found that GFOSS through this program of Google Summer of Code(GSoC) can help me become better in what I do and achieve all the goals that I am looking to achieve.

What am I doing before exact time of GSoC?

After the organisations that will participate in this year's GSoC were announced, I got excited in this project **MyUni** and I started contributing on the related project that was given so that I can start contributing to the existing project.

I have also started working on the initial figma design of some pages that will be part of the project . The link to the design is MyUni Initial Design

Timeline

Community Bonding Period (May 8 – June 1, 2025)

- Engage with **GFOSS mentors** and discuss the best implementation approach.
- Review **existing repositories** (MyUoM GitHub) and analyze system architecture.
- Understand more clearly intended solutions
- Gather **requirements** from potential university adopters.
- Finalize the design and wireframes of the system
- Create an **analysis and design document**, including:
    - System analysis
    - Feasibility study
    - Business procedures
    - User stories

- System backlog

Phase 1(June 8 - June 22)

Goal: Design the core backend system and frontend wireframes.

- **Set up a project repository** and initialize the backend (Express.js + MySQL).
- **Define the database schema** for users, announcements,notifications, and student data like reports,courses,performance and others.
- **Implement authentication and authorization** using JWT-based login.
- **Implement profile management** fot admins,students and teachers and all other users of the system
- **Re-writing existing codes in Typescript(TS)** for maintainability, scalability, improving code quality to ensure that the codes are bug-free.
- **Create wireframes and UI mockups** for the student dashboard and admin panel.
- **Submit initial design documentation.**
- **Work on student enrollment and course management** functionalities
- **Implement notifications:** A student will be able to get the notification based on announcements, events and other activities from their respective universities
- **Implement the UI** of pages concerning Authentication & Authorization,some pages that will be part of the student portal
- **Implement Multi-Tenancy support:** After this**,** System will be able to support many different institutions under one platform. For example, about the Database strategy, **Database Per Tenant** institution can be used.

- Write the documentation for future contributors

Phase 2 (June 23rd - July 7th)

Goal: Implement the core features.

- Develop a **custom CMS** to allow universities to manage announcements, schedules, and academic data.
- Implement **real-time synchronization** of data from university websites using **web scraping & APIs**.
- Build the **student portal** with personalized content, notifications and allow students get access to library resources so that student can borrow and return books conveniently.
- **Implement Internationalization** using i18n or Lingui so that the user can use comfortable language for him/her while using the system.
- Enable **multi-domain support** for multiple universities.
- Conduct **testing and performance optimizations**.
- Write documentation of codes written for future developers

July 9th - July 18th

Period of the special exam that I have to take from July 8th to July 18th

Phase 3 (July 19th - August 3rd)

- **Build the admin panel** for handling student information,manage their enrollments,other things

Finalize and deploy the project.

- Perform extensive **unit testing** and **integration testing**.
- Deploy the backend and frontend using **Docker & Kubernetes**.
- Document **setup instructions** for other universities.
- Finalize the project **repository** and submit for evaluation.

Deliverables:

- Web application that provide good user experience(UX) and has good User Interface(UI)
- Backend Architecture which serves as solution to all project requirements
- REST APIs with at least 80% test coverage
- REST APIs with modern design patterns

Technical Details

Technologies Used

- **Frontend:** React.js, Next.js (optional)
- **Backend:** Express.js, Node.js
- **Database:** MySQL
- **Authentication:** JWT-based authentication
- **Containerization:** Docker
- **Real-time Updates:** Web Scraping (Cheerio/Puppeteer) & API Integration
- **Multi-Domain Support:** vhost