# Project ID: 90

## Project Title

Building A Smart Assistant For Learning And Teaching

## Client Name

Basem Suleiman (B.Suleiman@Unsw.Edu.Au)

## Group Capacity

2 groups

## Project Background

As education becomes more digital, there is increasing interest in tools that can support students and educators in everyday learning tasks. This project focuses on building the foundation for a smart assistant that could eventually help with things like reviewing work, giving feedback, and answering student questions. The aim is to create a flexible web application that provides the basic structure for such a system. For now, the project is focused on building the core features needed to simulate how a digital assistant might work.

## Project Scope

Students will develop a full web application (frontend and backend) that includes key features to support learning environments. The platform should:

- Support multiple user roles, such as students and tutors
- Include features like login, user profiles, and dashboards
- Allow students to submit questions or assignments
- Allow tutors or the assistant to provide responses or feedback
- Simulate assistant responses (e.g. through mock data or basic AI tools)

The backend could be developed in Python, and the frontend can be built using a JavaScript-based framework. A chat-style interface is recommended and it can be used to simulate assistant interactions.

## Project Requirements

The application should include the following:

- User registration and login with role-based access
- Profile and dashboard pages for students and tutors
- A submission feature for content (e.g., questions, assignments)
- A response feature that can simulate or generate feedback
- Backend services that are modular and ready for future AI integration
- A clean, user-friendly frontend (e.g., chat-style UI)
- Optionally, use of tools like LangChain to simulate assistant behavior

## Required Skills

To succeed in this project, students should have experience with:
- Frontend development (HTML, CSS, and a framework like React or Vue)
- Backend development (preferably Python such as Flask, FastAPI, or Django)
- REST API design and implementation
- Basic understanding of how frontend and backend systems connect
- Interest in artificial intelligence and LLMs

## Expected Outcomes

By the end of the project, students should deliver:
- A working full-stack web application
- Source code with clear organisation and comments
- Technical documentation (e.g., system architecture, API specs)
- A basic user guide for using the application

## Disciplines

Web Application Development;Generative AI (GenAI);Human Computer Interaction (HCI);Cloud Computing;

## Other Resources

Will be provided in the kick-off meeting:
- Links to tutorials or documentation for OpenAI and LangChain
- Sample UI design references for inspiration (e.g., chat-based interfaces)
- Students may use free-tier cloud platforms or run the system locally, no paid services are required