COMP 019 - Applications Development and Emerging Technologies

Full-Stack Python Development

SESSION 1: INTRODUCTION TO FULL-STACK PYTHON DEVELOPMENT

ACTIVITY 1.1: PYTHON PROGRAMMING REVIEW

Topic: Core Python concepts for web development

Description: Review essential Python programming concepts needed for full-stack development

INSTRUCTIONS:

Complete Python fundamentals review:

- Data types: strings, lists, dictionaries, tuples, sets
- Control structures: if/else, for loops, while loops
- Functions: parameters, return values, *args, **kwargs
- Object-Oriented Programming: classes, inheritance, methods
- File handling: reading and writing files
- Exception handling: try/except blocks

Create a Python script that demonstrates:

- A class representing a 'Student' with attributes and methods
- Reading student data from a text file
- Storing students in a list and performing operations
- Writing results to an output file

Include proper error handling throughout

Deliverables: Submit Python script with Student class and file operations	**25 Points**

ACTIVITY 1.2: DEVELOPMENT ENVIRONMENT SETUP

Topic: Setting up professional Python development environment

Description: Configure your development environment for full-stack Python development

INSTRUCTIONS:

Install and configure required tools:

- Python 3.10+ (verify with python --version)
- VS Code with Python extension
- Git for version control

Set up virtual environment:

- Create: python -m venv venv
- Activate: venv\Scripts\activate (Windows) or source venv/bin/activate (Mac/Linux)
- Verify: pip list

Install essential packages:

- pip install django djangorestframework
- pip install kivy
- pip install psycopg2-binary
- Create requirements.txt: pip freeze > requirements.txt

Configure VS Code:

- Set Python interpreter to virtual environment
- Install recommended extensions (Pylance, GitLens)

Take screenshots of all successful installations

Deliverables: Submit screenshots of setup and requirements.txt file	**25 Points**

ACTIVITY 1.3: FULL-STACK ARCHITECTURE OVERVIEW

Topic: Understanding full-stack application architecture

Description: Learn the components and flow of a full-stack Python application

INSTRUCTIONS:

Research and document full-stack architecture:

- Frontend: User interface (HTML/CSS/JS or Mobile)
- Backend: Server-side logic (Django)
- Database: Data storage (SQLite/PostgreSQL)
- API: Communication layer (REST API)

Create architecture diagram showing:

- Client (Web browser or Mobile app)
- Web Server (Django)
- Database Server
- Request/Response flow

Document the technologies we'll use:

- Django for web backend
- Kivy for mobile frontend
- SQLite for local database
- PostgreSQL for cloud database
- Django REST Framework for API

Explain how each component connects to others

Deliverables: Submit architecture document with diagrams	**30 Points**

TOTAL POINTS: 80