COMP 019 - Applications Development and Emerging Technologies

Emerging Technologies: MCP, A2A, Pgvector

SESSION 9: MCP FUNDAMENTALS (EMERGING TECHNOLOGY)

ACTIVITY 9.1: INTRODUCTION TO MODEL CONTEXT PROTOCOL

Topic: Understanding MCP architecture and concepts

Description: Learn the fundamentals of Anthropic's Model Context Protocol

INSTRUCTIONS:

Research and document MCP:
- What is MCP? (Model Context Protocol)
- Created by Anthropic, now Linux Foundation standard
- Purpose: Standardize AI agent-to-tool communication
- Adopted by: OpenAI, Google, Microsoft

Understand MCP architecture:
- MCP Hosts (Claude Desktop, IDEs)
- MCP Clients (protocol connectors)
- MCP Servers (tool/data providers)
- Transport: stdio, HTTP+SSE

MCP primitives:
- Tools - Functions the AI can call
- Resources - Data the AI can access
- Prompts - Templates for common tasks

Compare MCP vs traditional API integration:
- Build once, connect to any AI
- Standardized interface
- Ecosystem benefits

Deliverables: Submit MCP research document with architecture diagrams	**25 Points**

ACTIVITY 9.2: BUILDING YOUR FIRST MCP SERVER

Topic: Creating an MCP server with Python

Description: Build a custom MCP server that exposes tools to AI agents

INSTRUCTIONS:

Set up MCP development environment:
- pip install mcp
- Understand FastMCP framework

Create simple MCP server:
- Initialize FastMCP server

• Define tool functions with decorators
• Add proper type hints and docstrings
Build 'Student Database' MCP server:
• Tool: get_student(student_id) - retrieve student info
• Tool: list_students(department) - list by department
• Tool: get_grades(student_id) - get student grades
• Tool: calculate_gpa(student_id) - calculate GPA
Connect to your Django database:
• MCP server reads from PostgreSQL
• Real data accessible to AI agents
Test with MCP Inspector:
• Run server locally
• Test each tool
• Verify responses

Deliverables: Submit MCP server code and Inspector test screenshots	**40 Points**

ACTIVITY 9.3: INTEGRATING MCP WITH APPLICATIONS

Topic: Using MCP servers in applications

Description: Connect your MCP server to AI applications and test integration

INSTRUCTIONS:

Configure Claude Desktop (if available):
• Add MCP server to claude_desktop_config.json
• Restart Claude Desktop
• Test tools in conversation
Build MCP client in Python:
• Install mcp client library
• Connect to your MCP server
• List available tools programmatically
• Call tools and process results
Create practical application:
• Python script that uses MCP client
• Natural language interface for database queries
• 'Find all students in IT department'
• 'What is the GPA of student 12345?'
• MCP translates to tool calls
Document the integration flow
Discuss MCP benefits in your application

Deliverables: Submit MCP client code and integration demo	**30 Points**

TOTAL POINTS: 95