COMP 019 - Applications Development and Emerging Technologies

Project Development Phase

SESSION 13: PROJECT SPRINT 1 - FOUNDATION

ACTIVITY 13.1: PROJECT PROPOSAL AND REQUIREMENTS

Topic: Defining your capstone project

Description: Create comprehensive project proposal incorporating all technologies

INSTRUCTIONS:

Choose project idea (or propose your own):

- AI-Powered Learning Management System
- Smart Inventory Management with Predictions
- Intelligent Document Q&A System
- AI Research Assistant
- Smart Customer Service Platform

Document requirements:

- Problem statement and target users
- Functional requirements (minimum 8 features)
- Technical requirements:
  - Django backend with REST API
  - PostgreSQL with Pgvector
  - MCP server (minimum 3 tools)
  - A2A integration (optional but bonus)
  - RAG for intelligent search
  - Mobile or Web frontend

Create project timeline (4 sprints)

Get instructor approval

Deliverables: Submit project proposal document	**30 Points**

ACTIVITY 13.2: DJANGO BACKEND SETUP

Topic: Building the core backend infrastructure

Description: Set up Django project with database models and basic API

INSTRUCTIONS:

Create Django project:

- Set up project structure
- Configure PostgreSQL with Pgvector
- Create all required models
- Set up Django admin

Design database schema:

• Main application tables
• Vector-enabled tables for RAG
• Relationships and constraints

Create basic REST API:

• Serializers for all models
• ViewSets for CRUD operations
• URL routing
• Basic authentication

Initial data setup:

• Create migrations
• Seed with sample data
• Generate embeddings for vector content

Test all endpoints with Postman

Deliverables: Submit Django project code and API test results	**40 Points**

ACTIVITY 13.3: FRONTEND FOUNDATION

Topic: Building the user interface foundation

Description: Create the basic frontend structure for your application

INSTRUCTIONS:

Choose frontend approach:

• Kivy mobile app, OR
• Web frontend (Django templates or React)

Create UI structure:

• Authentication screens (login/register)
• Main dashboard/home screen
• List views for main data
• Detail views
• Forms for data entry

Connect to backend API:

• API client setup
• Authentication flow
• Basic CRUD operations working

UI/UX considerations:

• Consistent styling
• Loading states
• Error handling
• Responsive design (if web)

Milestone: 25% complete

Deliverables: Submit frontend code and working demo (25% completion)	**35 Points**

TOTAL POINTS: 105