COMP 019 - Applications Development and Emerging Technologies

Full-Stack Python Development

SESSION 6: CLOUD DATABASE WITH POSTGRESQL

ACTIVITY 6.1: POSTGRESQL SETUP AND BASICS

Topic: Setting up PostgreSQL for production use

Description: Install and configure PostgreSQL database

INSTRUCTIONS:

Option A - Local PostgreSQL:
- Download and install PostgreSQL
- Create database and user
- Note connection details

Option B - Cloud PostgreSQL (Recommended):
- Create account on Supabase, Neon, or Railway
- Create new PostgreSQL database
- Get connection string

Test connection:
- Install: pip install psycopg2-binary
- Connect with Python script
- Create test table and insert data

PostgreSQL vs SQLite:
- Multi-user support
- Better performance at scale
- Advanced features (JSON, full-text search)
- Required for production deployment

Deliverables: Submit PostgreSQL setup proof and connection test screenshots	**25 Points**

ACTIVITY 6.2: MIGRATING DJANGO TO POSTGRESQL

Topic: Configuring Django to use PostgreSQL

Description: Switch your Django project from SQLite to PostgreSQL

INSTRUCTIONS:

Update Django settings:
- Modify DATABASES in settings.py
- Use dj-database-url for cleaner config
- Store credentials in environment variables

Database configuration:

DATABASES = {

```
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'your_db_name',
        'USER': 'your_user',
        'PASSWORD': 'your_password',
        'HOST': 'your_host',
        'PORT': '5432',
    }
}
```

Migrate data:

• python manage.py migrate

• Create new superuser

• Verify in admin panel

Optional: Export SQLite data and import to PostgreSQL

Deliverables: Submit settings.py (credentials hidden) and Django admin connected to PostgreSQL	**30 Points**

ACTIVITY 6.3: ENVIRONMENT VARIABLES AND SECURITY

Topic: Securing database credentials and configuration

Description: Implement secure configuration management for your application

INSTRUCTIONS:

Install python-dotenv:

• pip install python-dotenv

Create .env file:

• DATABASE_URL=postgres://user:pass@host:5432/db

• SECRET_KEY=your-secret-key

• DEBUG=False

Update settings.py:

• from dotenv import load_dotenv

• load_dotenv()

• SECRET_KEY = os.getenv('SECRET_KEY')

• Use dj_database_url.config()

Security practices:

• Add .env to .gitignore (NEVER commit credentials)

• Use different credentials for dev/prod

• Rotate credentials periodically

Create .env.example:

• Template showing required variables

• No actual values, just placeholders

• Commit this file to help other developers

Deliverables: Submit settings.py, .env.example, and .gitignore	**30 Points**

TOTAL POINTS: 85