Capstone Tasks - Theoretical Explanation

1. Build a pipeline that runs expense analysis weekly or monthly

- Store the expense analysis script (Python / PySpark) in Azure Repos (or GitHub).
- Create a YAML pipeline in Azure DevOps that:
    - Installs dependencies (Python, PySpark if needed).
    - Pulls the latest cleaned data from source (CSV/Delta in storage).
    - Run the analysis script.
- Use scheduling triggers in the pipeline:
    - cron style schedule → every Sunday night (weekly) or 1st day of month.

2. Output summary report as CSV

- The pipeline saves results (monthly spend, savings, alerts) into a CSV.
- CSV is published as a pipeline artifact, so it can be downloaded.
- Alternatively, it can be pushed to Azure Storage (Blob/ADLS).

3. Log or print a savings alert if expenses exceed threshold

- Script checks if spending > 80% (or any set threshold) of monthly income.
- If exceeded → print a warning message → pipeline logs capture it.
- Could also send an Azure DevOps notification or email alert via extensions.

Steps to Create CI/CD Pipeline in Azure DevOps

1. Create a Project in Azure DevOps

- Go to dev.azure.com → sign in.
- Click New Project → give a name (like ExpenseManagement).
- Choose Public/Private → click Create.

2. Push Your Code to Repos

- Go to Repos → copy the Git URL.
- Clone locally:

```
git clone <my_repo_url>
```

- Add your PySpark/SQL scripts

3. Create a New Pipeline

- Go to Pipelines → Create Pipeline.
- Choose Azure Repos Git (or GitHub if stored there).
- Select the repo.
- Choose Starter Pipeline or YAML file.

4. Define Pipeline Stages in YAML

5. Run the Pipeline

- Save the YAML → commit to repo.
- Go to Pipelines → Run Pipeline → select branch → Run.
- Azure DevOps executes each stage step by step.