Capstone Tasks – Theoretical Explanation

1. **Automate the energy report generation weekly**
    - Set up a CI/CD pipeline in Azure DevOps Pipelines.
    - Schedule the pipeline to run once a week (using cron triggers or scheduled runs).
    - The pipeline will trigger scripts (PySpark/SQL) that read the energy data, generate weekly summary reports, and store them as CSV/Delta files.

2. **Pipeline to fetch new data, clean, and summarize**
    - The pipeline has sequential stages:
        - Fetch Stage - Pull new device data from storage (e.g., Azure Blob or SQL DB).
        - Clean Stage - Apply data cleaning scripts (remove nulls, duplicates).
        - Summarize Stage - Run aggregation jobs to compute daily/weekly summaries (avg, peak/off-peak, per device).
    - Each stage runs automatically when triggered, ensuring continuous integration of new data.

3. **Alert or log if usage crosses a threshold (>10 kWh per device per day)**
    - Add a step in the pipeline to check thresholds after summarization.
    - If any device exceeds 10 kWh/day, log a warning in Azure DevOps logs or trigger an Azure Monitor alert.
    - Alerts can notify via email, Teams, or dashboards, helping track abnormal consumption.

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 (e.g., EnergyAutomation).
- 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 (data fetch, clean, summarize, alert check).

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.