Week 5 – CI/CD Pipeline with Azure DevOps

Tools Used

- Azure DevOps

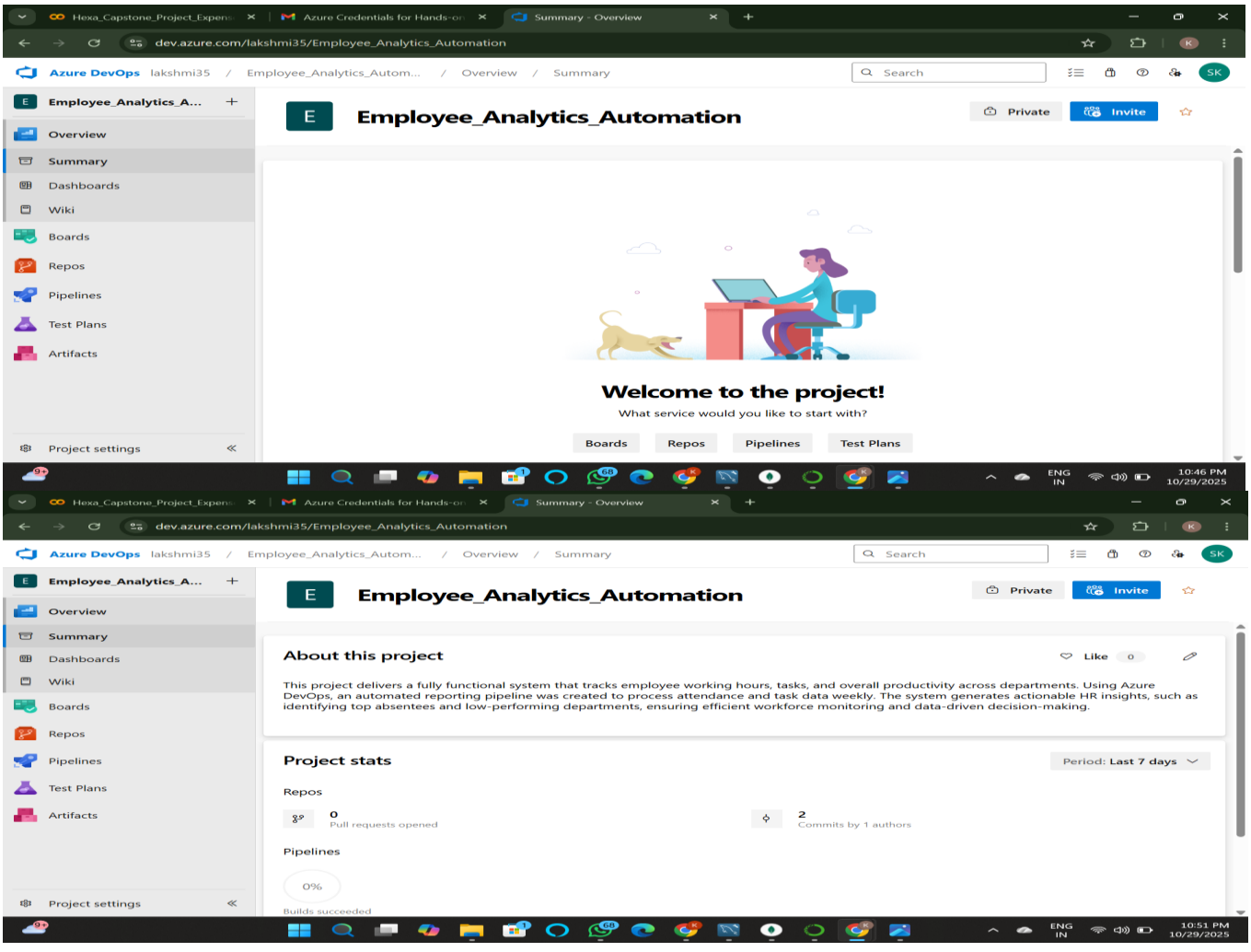
Objective

- To automate the Employee Attendance and Productivity Tracker system using a CI/CD pipeline in Azure DevOps.
- The goal was to schedule weekly execution of the ETL and reporting scripts to generate insights such as top absentees and low-performing departments, minimizing manual effort.

Steps Followed

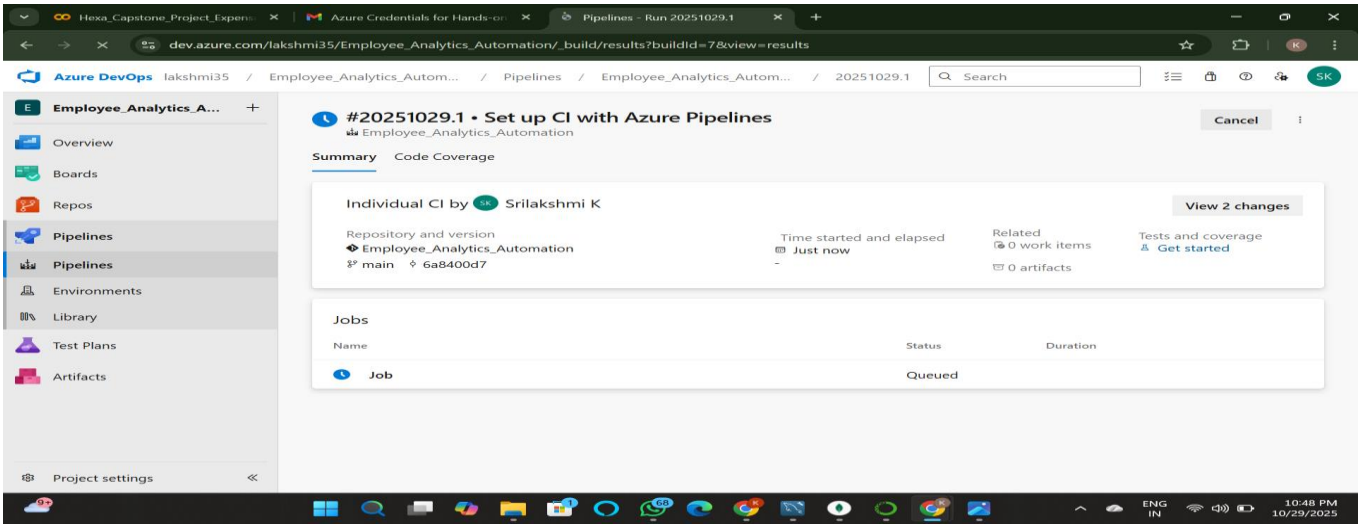
Step 1: Created a New Project

- Opened Azure DevOps.
- Created a new project named **Employee_Analytics_Automation**.
- Selected Public visibility for easier configuration and testing.



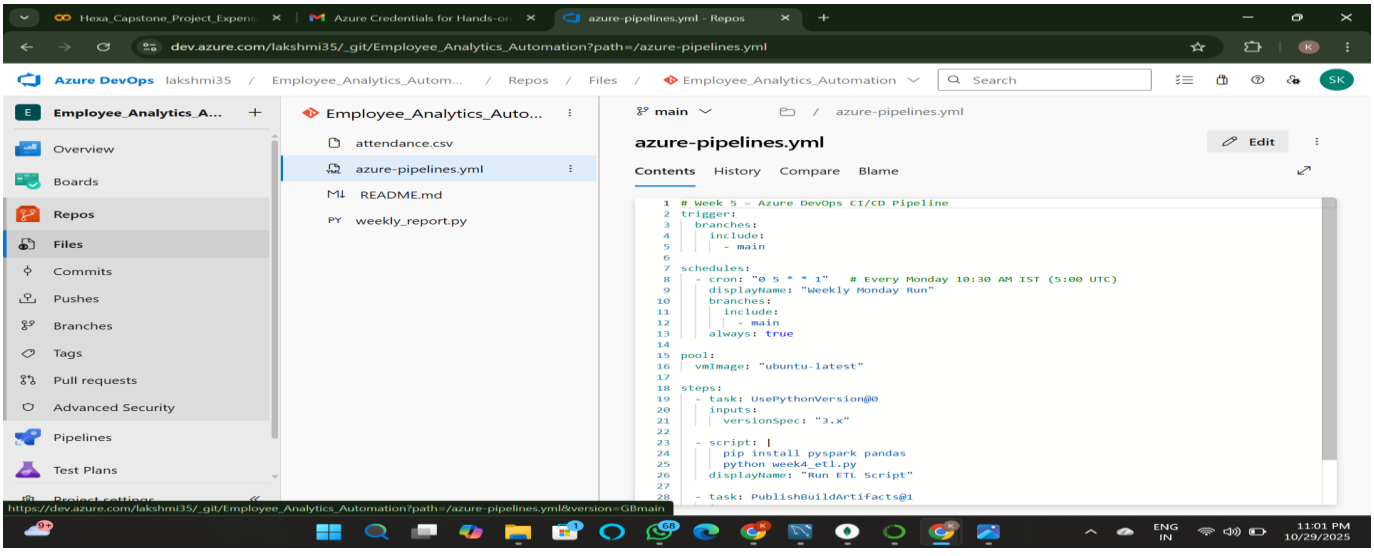
Step 2: Created a New Pipeline

- Navigated to **Pipelines → Create Pipeline**.
- Selected the **Starter pipeline** option to begin from a basic template.
- Used the YAML editor to define automation steps.



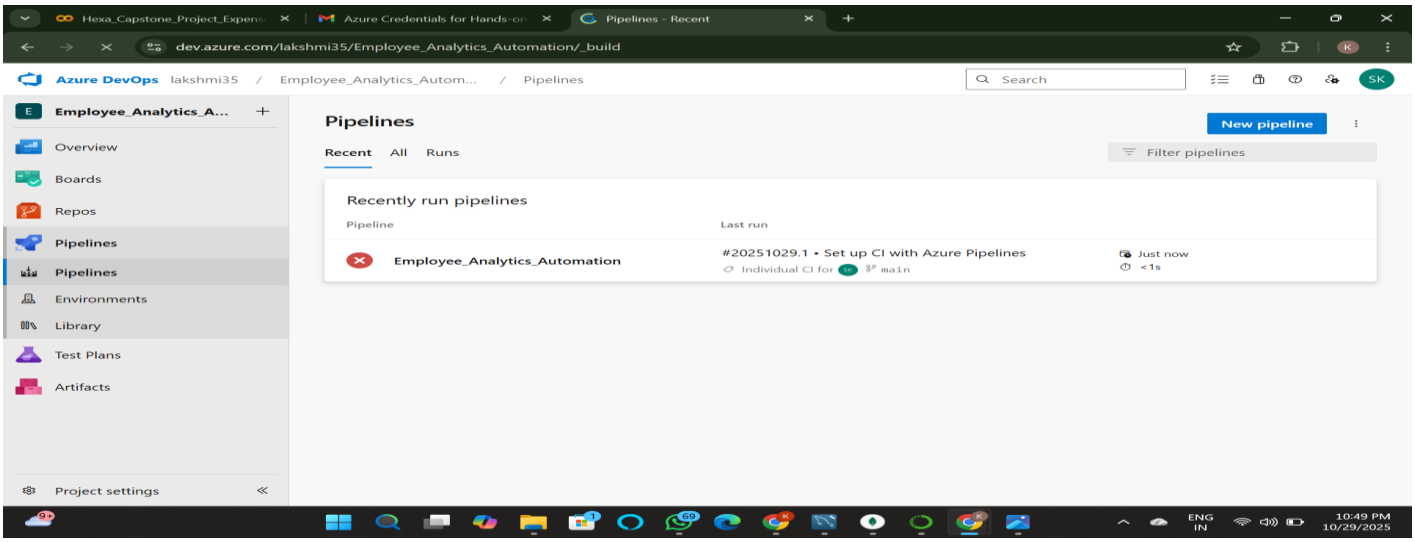
Step 3: Added YAML Configuration

- Wrote a sample YAML configuration file that installs dependencies (like pandas, numpy) and runs the attendance tracking script.
- Saved the configuration as **azure-pipelines.yml**.



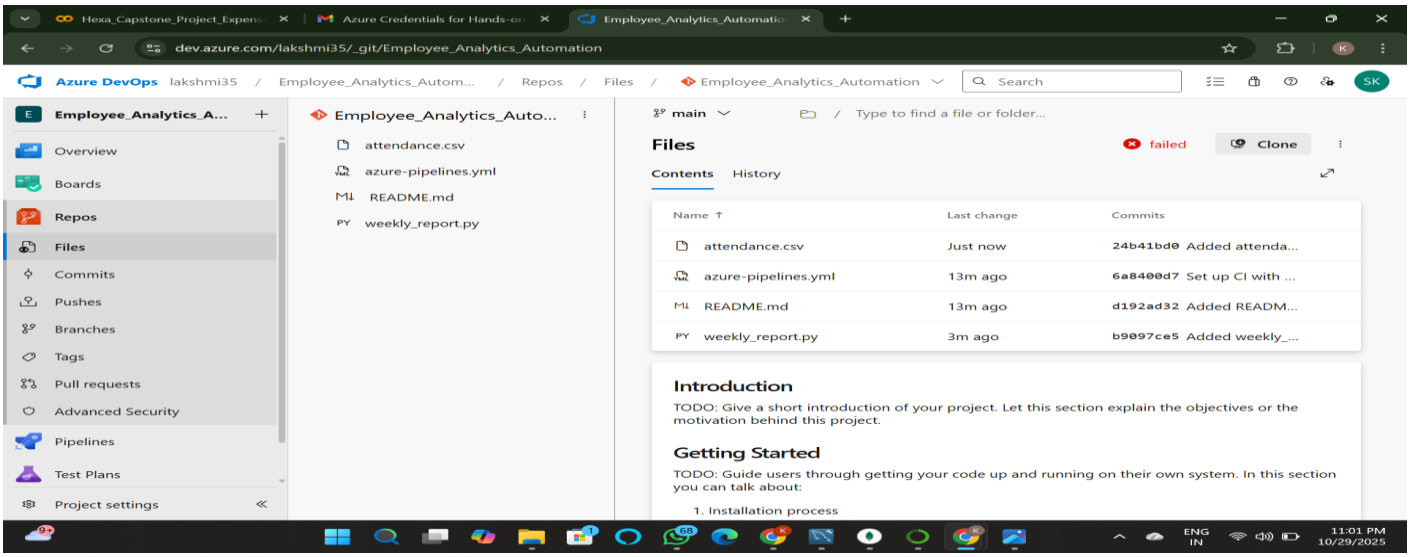
Step 4: Ran the Pipeline

- Clicked on **Run Pipeline** to execute the process.
- The run started successfully, but execution stopped due to hosted parallelism settings.
- Verified that the pipeline setup and YAML configuration were correct.



Deliverables

- YAML pipeline configuration file
- Screenshots of each step ,Project creation, Pipeline setup, YAML code, Execution



Conclusion

The CI/CD pipeline setup in Azure DevOps was completed successfully at the configuration level.

Although the execution didn't produce artifacts due to hosted agent limitations, the configuration demonstrates a clear understanding of CI/CD automation for the Employee Attendance Tracker project.