Week 5 – CI/CD Pipeline with Azure DevOps

Tools Used

- Azure DevOps

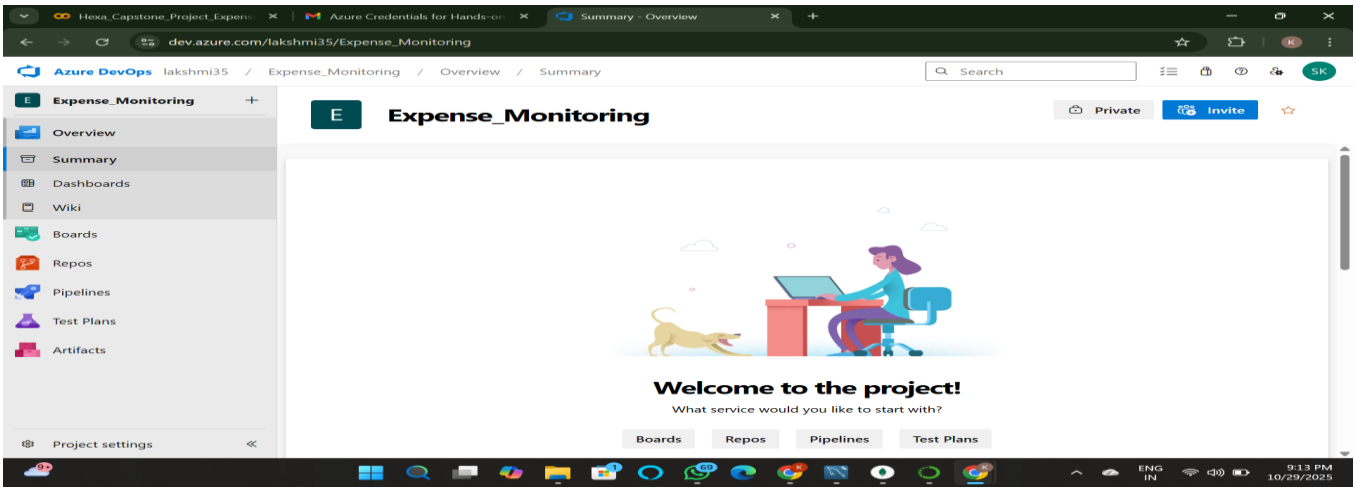
Objective

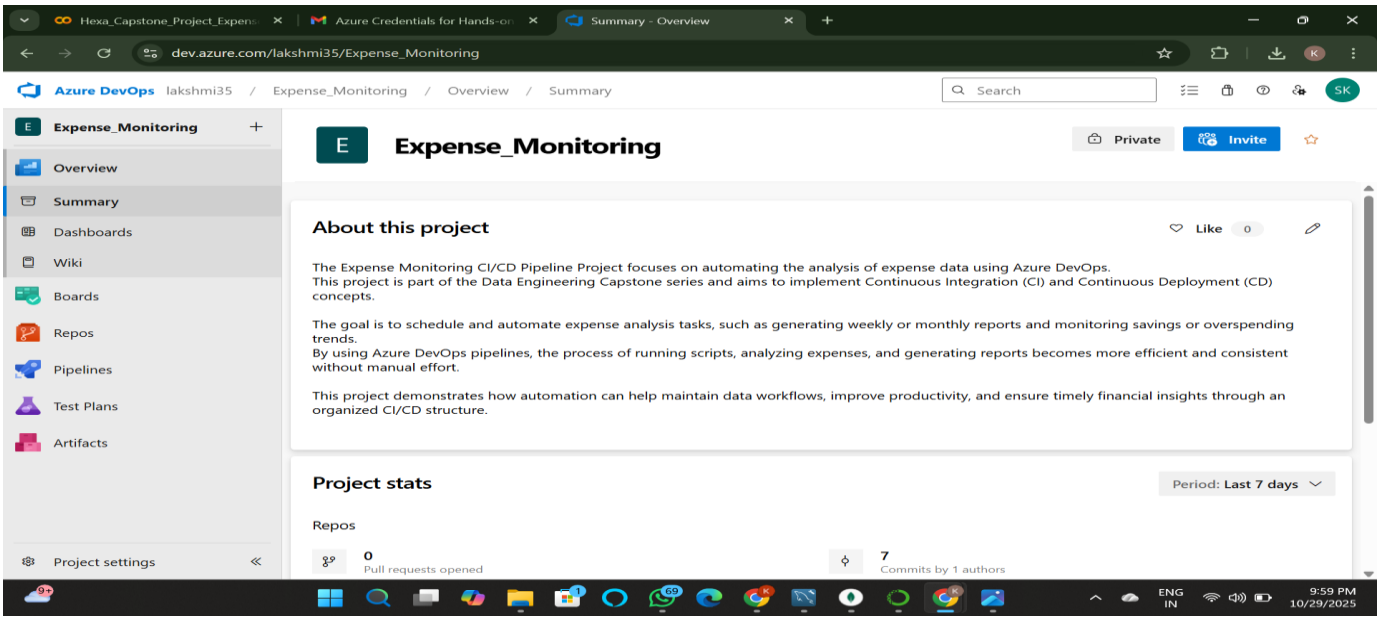
To create a CI/CD pipeline that automates the weekly expense analysis process using Azure DevOps.

Steps Followed

Step 1: Created a New Project

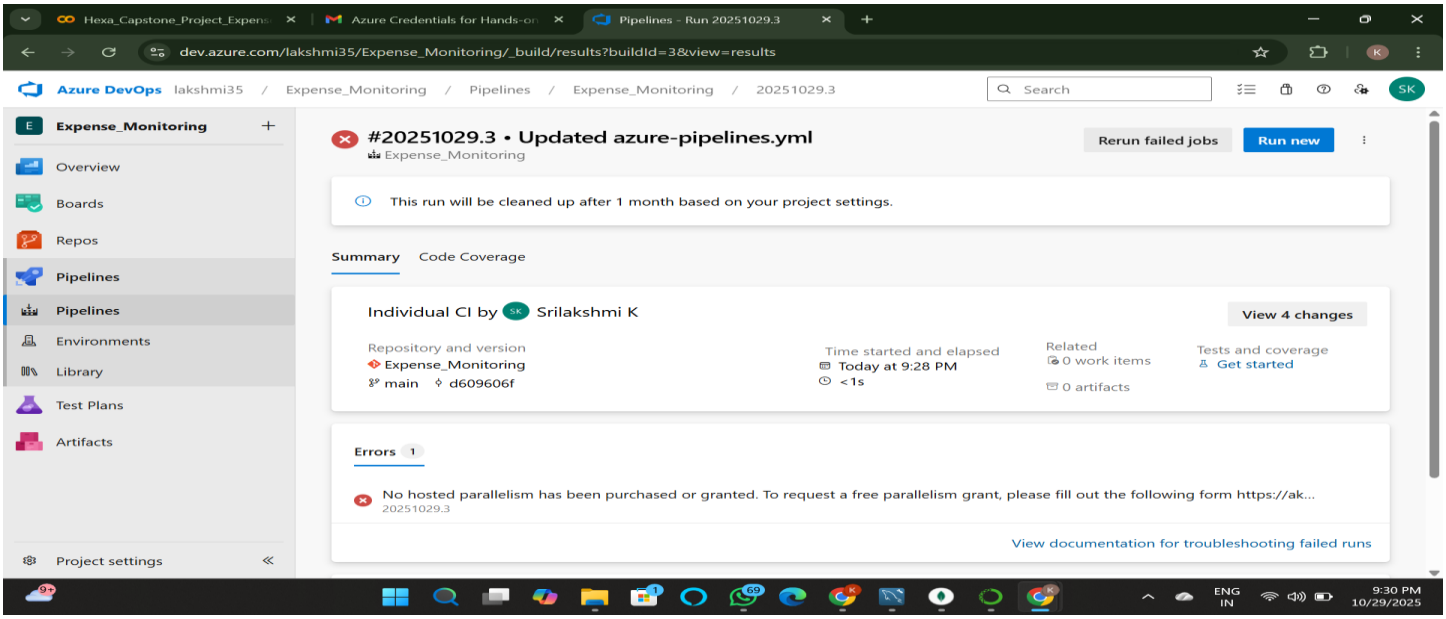
- Opened Azure DevOps.
- Created a new project named **Expense_Monitoring**.
- Selected *Public visibility* to make setup easier.





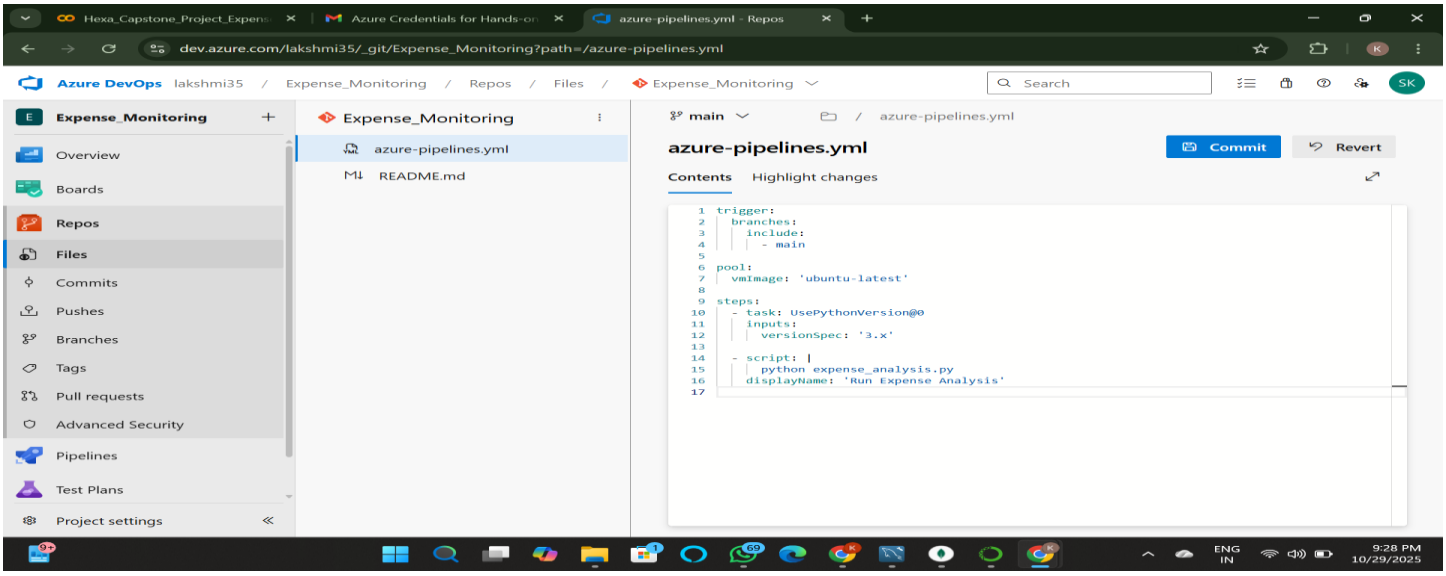
Step 2: Created a New Pipeline

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



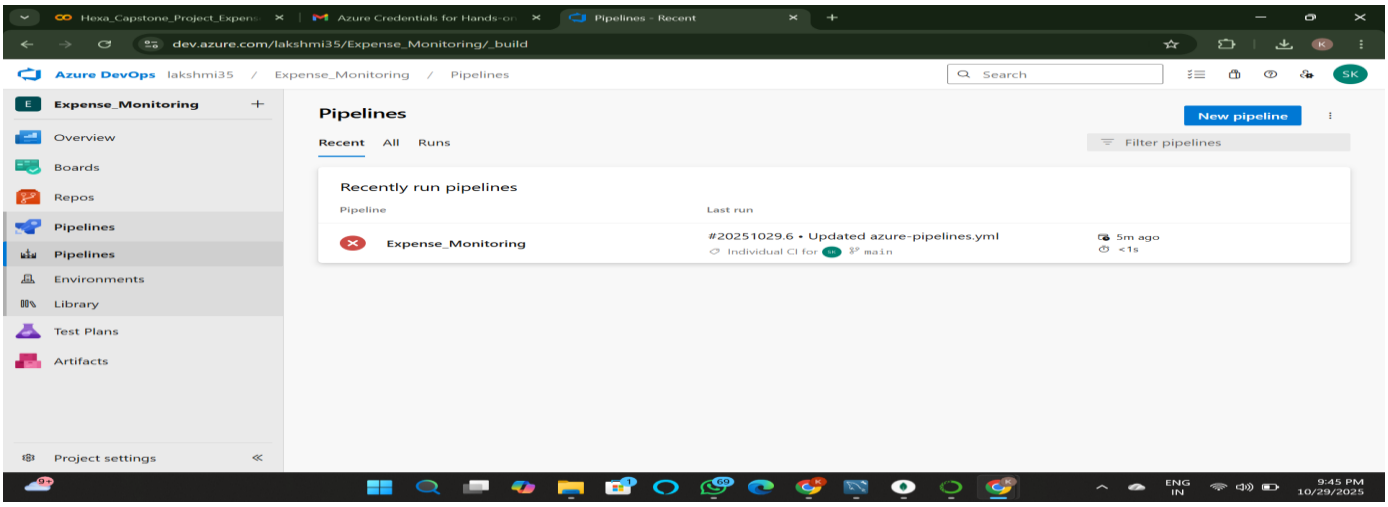
Step 3: Added YAML Configuration

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



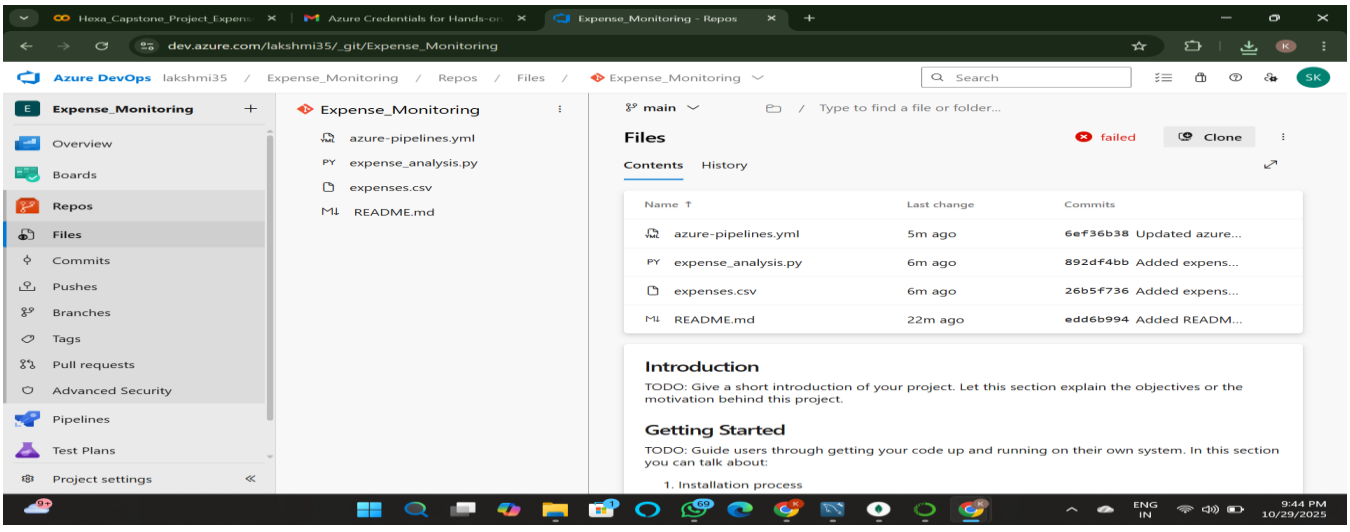
Step 4: Ran the Pipeline

- Clicked on **Run Pipeline** to execute the process.
- The run started successfully, but execution stopped due to hosted parallelism settings.
- Observed logs to identify that execution setup was 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 execution didn't produce artifacts due to hosted agent limitations, the pipeline design and YAML structure demonstrate a clear understanding of CI/CD automation setup.