**Shape, rectangle

Description automatically generated**



**AI Companion – POC**

**Oct 2025**

**Version 1.0**

Table of Contents

[1.1 Objective 3](#_Toc210737663)

[1.2 Team & Timeline 3](#_Toc210737664)

[1.3 AI Companions leveraged 3](#_Toc210737665)

[1.4 Environment Setup 3](#_Toc210737666)

[1.5 Existing Code Structure and Setup 4](#_Toc210737667)

[1.6 Scenarios & Outcomes 5](#_Toc210737668)

## Objective

To explore the capability of various AI Companions tools to handle complex enhancements on large, highly customizable bespoke applications and demonstrate the developer productivity gained through their adoption.

## Team & Timeline

|  |  |
| --- | --- |
| **The Team** | |
| **Name** | **Role** |
| Subhash Bhaskaran | AI Engineer |
| Murugan Sundaram | Data Hub SME |
| Viswanatha R | Data Hub SME |

**Start Date: 1st Oct 2025**

**End Date: 10th Oct 2025**

## AI Companions leveraged

1. Devin
2. Co Pilot

## Environment Setup

1. Developer Laptop
   * Github
   * Any IDE like IntelliJ or VS Code with Co Pilot plugin enabled
   * Devin web subscription
2. Code
   * Python based ingestion framework
   * Any data, excel reconciliation code from public domain, cloned locally

## Existing Code Structure and Setup

**GitHub Location**

https://github.com/ltibfspoc/devinpoc

**Summary**

The code is data processing ETL application built using PySpark for distributed data transformation and Luigi for workflow orchestration. The application supports transformation like filter, sort, merge, etc. The application is configurable using json template to define data pipeline.

Devin was then asked to understand the code structure and build its knowledge. Devin using this knowledge then was asked to work on different scenarios.

**Structure**

A screenshot of a computer

AI-generated content may be incorrect.

## Scenarios & Outcomes

|  |  |
| --- | --- |
| **Scenario 1** | Requirement to create a new ingestion job using an existing source file.  In the Prompt provide the source file name and the name of the column to be used and the aggregation logic. Also provide the target table name  Tool Plan Evaluation:   * able to identify all the necessary changes to the job and configuration files * able to identify the transformation logic changes in configuration files * ability to identify existing ingestion job(s), that can be used as base for this new requirement * able to write unit test cases and pass them |
| **Prompt** | Using the Test1 branch perform the below new requirement  1) Create a new pipeline by name "test51.json" to read the csv file "factbook.csv" from the "data" folder.  2) After reading the file "factbook.cs", filter the records with condition "Area < 300"  3) Put the output in folder "factbook300" in the folder "outputfile".  4) Run the ingestion job by changing the run\_pipeline.py |
| **AI Companions Used** | Devin |
| **Outcome** | Devin was able to understand the configuration required to develop the new ingestion job and ensure only the relevant code change was performed. |
| **Evidence** |  |

|  |  |
| --- | --- |
| **Scenario 2** | Requirement to create new test cases based on the requirement and code generated for scenario 2  In the Prompt provide requirements of the new ingestion job  Tool Plan Evaluation:   * able to identify all the necessary scenarios for which testing is required (Happy Path, Edge Case, Boundary Condition) * Ability to create plain text Test Case, with the steps to be performed to execute and validate the Test Case. Test Case should follow the format that can be copy paste into Zypher |
| **Prompt** | Using the Test1 branch performs the below new requirement  1) Write new python code to test the pipeline “test51.json” ingestion job  2) create the new code in a new folder called “testautomation” under “src”  3) The new code should take the source file and output file path and filename as parameter  4) Test Case scenario 1: The new code should verify the row count match between source and target file based on the filter condition  5) Test Case scenario 2: The new code should verify if all records in output file have Area < 300  6) Test Case scenario 3: The new code should verify the sum of CurrentAccountBalance match between source and target file based on the filter condition  7) The new code should create an excel file called “result51” in a new folder called result" under "src" containing the results of this test. Sheet 1 is summary, Sheet 2 should indicate each column tested for count or sum depending on the data type. |
| **AI Companions Used** | Devin |
| **Outcome** | Devin was able to create the test automation script and create the result in the format specified. |
| **Evidence** |  |

|  |  |
| --- | --- |
| **Scenario 3** | Generating Technical Documentation  In the Prompt provide the location of the Github and the describe the format of the Documentation  Tool Plan Evaluation:   * able to incorporate all critical points from codebase * able to summarize in easy-to-understand language for developers * Documentation should follow the format that can be copy paste as readme file. |
| **Prompt** | try harder to access the "https://github.com/ltibfspoc" and repo name is "devinpoc" and provide me the documentation and steps to run this python code |
| **AI Companions Used** | ChatGPT enabled as Agent mode and provided access to Github repository |
| **Outcome** | Read the GitHub repository and complied content in concise format with tables and diagram. |
| **Evidence** |  |

|  |  |
| --- | --- |
| **Scenario 4** | Requirement to add a new column to an ingestion job since the source file has new column.  In the Prompt provide the source file name and the name of the new column to be added.  Tool Plan Evaluation:   * able to correctly identify the correct ingestion job * able to identify all the necessary changes to the job and configuration files * able to identify the target table(s) and the changes to be made * able to write unit test cases and pass them |
| **Prompt** |  |
| **AI Companions Used** |  |
| **Outcome** |  |
| **Evidence** |  |
| **Summary** |  |

|  |  |
| --- | --- |
| **Scenario 5** | Requirement to create a new test automation code based on the test cases generated in scenario 3.  In the Prompt detailed test case steps and validation  Tool Plan Evaluation:   * able to identify all the necessary changes to the job and configuration files * ability to identify existing test case code, that can be used as base for this new test automation code * able to execute the test automation code and pass them |
| **Prompt** |  |
| **AI Companions Used** |  |
| **Outcome** |  |
| **Evidence** |  |

|  |  |
| --- | --- |
| **Scenario 6** | Performance Improvement or Refactoring of existing code  In the Prompt provide ingestion job that needs refactoring  Tool Plan Evaluation:   * able to identify all the changes to the ingestion job * Ability to explain how those changes will improve performance |
| **Prompt** |  |
| **AI Companions Used** |  |
| **Outcome** |  |
| **Evidence** |  |

|  |  |
| --- | --- |
| **Scenario 7** | Generating Business Requirement documentation  In the Prompt provide meeting transcription and other details  Tool Plan Evaluation:   * able to incorporate all critical points from meeting * able to summarize in easy-to-understand language for developers * Requirement should follow the format that can be copy paste into Jira as stories. |
| **Prompt** |  |
| **AI Companions Used** |  |
| **Outcome** |  |
| **Evidence** |  |

