**Betsson Group Customer Service Automation**

**Project Modules Explanation with screenshots**

1. **Configuration :** in this module all initial variables which are needed for project are assigned. Relative file path: “./Data/Config.xlsx”

Changes which needs to done:

1. OpenaiAPI needs to be created and paste the API key in respective cell.
2. GmailClientID needs to be created and paste the ID in respective cell
3. GmailClientSecret needs to be created and paste the secret in respective cell.
4. PythonPath, absolute path of python should be specified.
5. PythonLibrary, absolute path of python library should be given in cell
6. PartOfProject, which part of Project requirement to run *(value 1 for first part with out AI integration and value 2 for AI integration)*

Note: all the references are given in README.md file

1. **DataRetrivelFromForms :**
2. Static forms are created in project folder named Forms.

**A screenshot of a computer

AI-generated content may be incorrect.**

1. Simple form with fields Name, Email, Issue Category and Issue Description

**A screenshot of a computer

AI-generated content may be incorrect.**

1. Initial scraping is done, and data is stored in file “./Data/CustomerData.xlsx” with unique case ID.
2. In this process data retrieval is done looping folder and going through web page and scraping data.
3. All eventlog is stored in “./log/DataScrappingStatuslog.Xlsx”
4. All system exceptions are stored in “./log/Exception.txt”
5. **Data Validation:**
6. In this module we loop the “./Data/CustomerData.xlsx” excel file into Data Table
7. Check in each row (this means each customer details) Email format and name format with regex and if it founds any wrong format it will update in remarks column.
8. Finally store the validated table into “./Data/ValidatedCustomerData.Xlsx”
9. All eventlog is stored in “./log/DataScrappingStatuslog.Xlsx”
10. All system exceptions are stored in “./log/Exception.txt”

*Note: if we want to run Initial flow (first part of requirements) keep the flag in Config.xlsx file in Column Name “PartOfProject” if value is 1 it will send Emails with only Case ID.*

*And stops the process, if value in sheet is 2 then it will run Python script for subcatogries and send email with sub category.*

1. **SendEmailToCustomer(for first part of Process):**
2. If *PartOfProject* 1, Read “./ValidatedCustomerData.xlsx” excel file and store it in Data Table, filter out customer rows whose name or email format is wrong
3. Loop the Data Table and send email with CaseID and message
4. Body is created in HTML so there are place holders to send Dynamically.
5. For this Gsuite setup is also required

Note: Gsuite setup document is also provided in README.md

1. All eventlog is stored in “./log/DataScrappingStatuslog.Xlsx”

A screenshot of a computer

AI-generated content may be incorrect.

1. **SubcategoriesDecision:** 
   1. This is the updated process where AI modules is called via OpenAi LLM API.
   2. This is developed in python script in path “./PyScripts/Subcategories.py”
   3. Parameters which are sent to this script are, issue description, issue category and openAi api key.
   4. Response is again string manipulated such a way that it will only return two to three words, if it cannot analyze with parameters which we have passed it will return “Sorry cannot suggest Sub category”
   5. This is all stored in file “./Data/AiCustomerData.xlsx”
2. **SendEmailToCustomer(for second part of Process):**
3. Read “./AICustomerData.xlsx” excel file and store it in Data Table, filter out customer rows whose name or email format is wrong
4. Loop the Data Table and send email with CaseID and message with subcategory.
5. Body is created in HTML so there are place holders to send Dynamically.
6. For this Gsuite setup is also required

Note: Gsuite setup document is also provided in README.md

1. All eventlog is stored in “./log/DataScrappingStatuslog.Xlsx”
2. All system exceptions are stored in “./log/Exception.txt”

A screenshot of a computer screen

AI-generated content may be incorrect.

1. **Additional Emails to customer:** 
   1. This module also calls same module SendEmailToCustomer uses.
   2. But an Argument is passed such a way that it will trigger another email which asks customer to submit some more description regarding issue they have faced.
   3. It loops AiCustomerData and filters sub category column which contains “sorry” in it and then stores in data table and loop the filtered table and proceed with emails to customer.
2. **Dashboard using PowerBI**:
   1. Dashboard is created with two graphs
      1. Bargraph for number of requests for each category.
      2. Piechar for percentage of request for each category.
   2. For this goto file path “./Data/Dashboard.pbix” and first give absolute path of file “./Data/ValidatedCustomerData.xlsx” in your directory because PowerBi do not support relative file paths.
   3. Goto File > options & settings > Data Source settings

A screenshot of a computer

AI-generated content may be incorrect.

* 1. After this refresh from Data Pannel as shown in below screenshot.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

* 1. Output looks like. A screenshot of a graph

     AI-generated content may be incorrect.