**## What is the project about? Problem it wants to solve? Goals?**

This chatbot will assist employees by providing instant information about various insurance coverages during customer interactions.

particularly in understanding and applying the different insurance coverages.

* ***Goals***\*: Enhance Employee Efficiency, Improve Customer Service, Streamline Operations

**## Does the project work? Does the code run? Is the output that one expected?**

good answer comes out based on the pdf's content. (page 3,4)

**## How is the project organised? Building blocks? What does what?**

The key file is methods and Support

1. [methods.py](http://methods.py): Contains the core methods for processing PDFs, creating vector stores, managing the chatbot's conversational chains, validatig answers, uploading and updating documents on mongodb
2. [Support.py](http://Support.py): user interface for interacting with the chatbot and a checking system health button.
3. [Reporting.py](http://Reporting.py): Generates reports based on user questions and sends emails to highlight areas where training is needed.
4. [mongotest.py](http://mongotest.py): A script to test the MongoDB connection.
5. 1\_Validierung.py: A page in my app for validating chatbot responses with the help of experienced employees.

**## If it not working: Why? Is it a code problem or an approach problem? How to fix it?**

* ***cannot send the email***\* [Reporting.py](http://Reporting.py):

- \*\*cannot send the email\*\* Reporting.py:

smtplib.SMTPAuthenticationError: (535, b'5.7.8 Username and Password not accepted. For more information, go to\n5.7.8 https://support.google.com/mail/?p=BadCredentials ffacd0b85a97d-36075093a34sm3215793f8f.5 - gsmtp')

- I use different email in EMAIL\_SENDER

- soultion:

The email should be activated a the 2-step verification

and then create an app password

**## If It is working: How is the performance? How to improve it (concrete suggestions)?**

* As the developer mentioned, everytime reloading the application, the (pdf's file)vector\_store is reloaded and store in the database
* => get\_vector\_store in [methods.py](http://methods.py):
* Checking for Existing Vector Store: In get\_vector\_store, the code checks if the persist\_directory exists. If it does, it loads the existing vector store instead of creating a new one.
* get\_systems\_health in [methods.py](http://methods.py):
* it would be nice if there was a function that tells which part the chatgpt referred to and answered