1. Create a Flask application with an /api route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

Ans:- The code

from flask import Flask, request

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

    return 'Welcome to the Home Page!'

@app.route('/api')

def name():

    name= request.values.get('name')

    email = request.values.get('email')

    result = {

        'name': name,

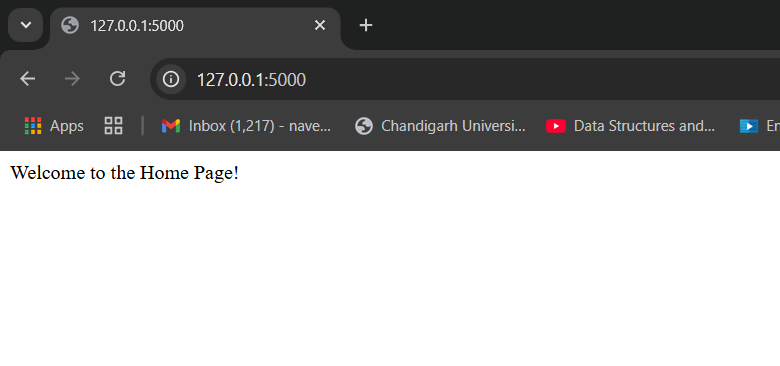
        'email': email

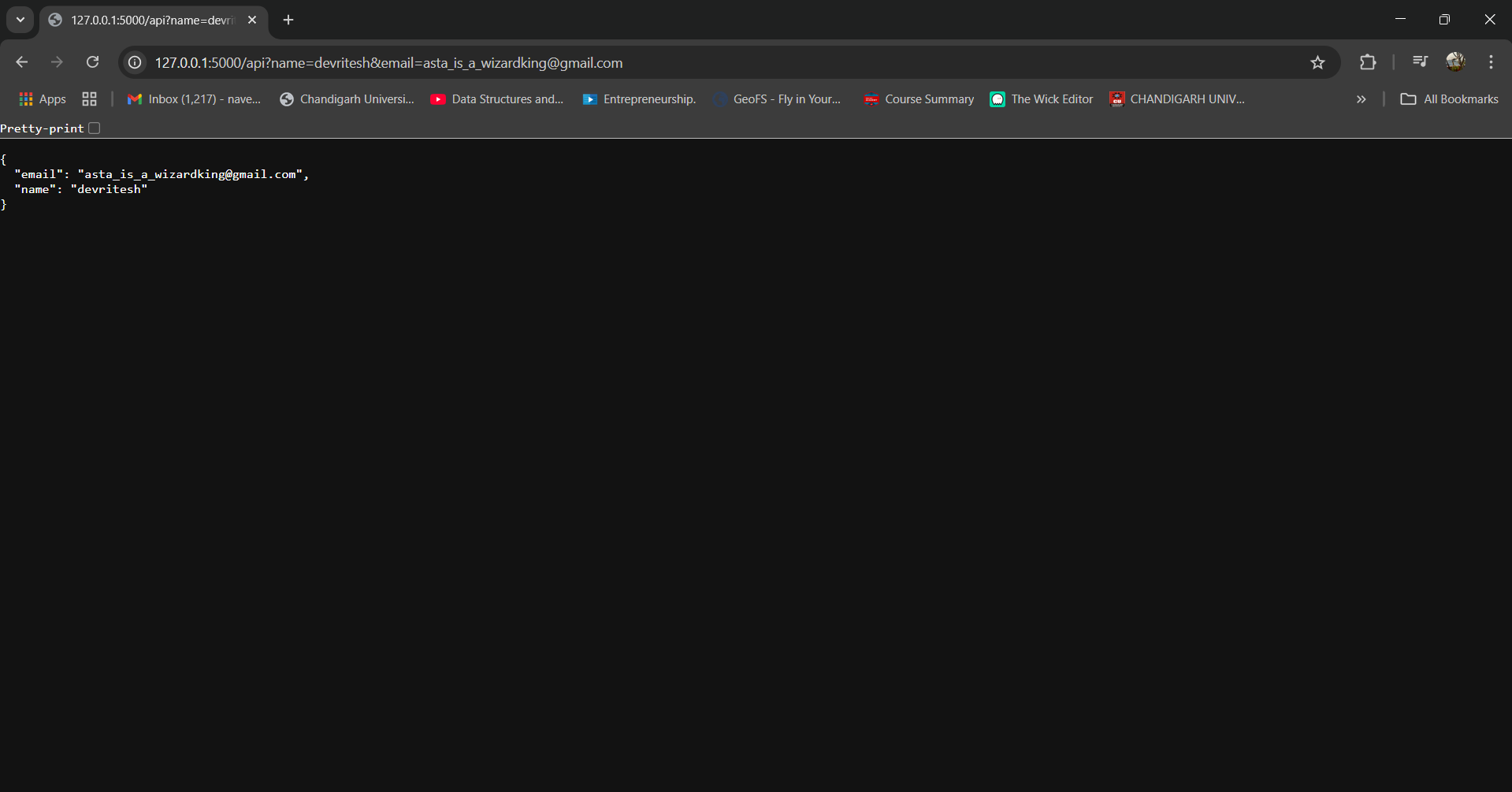
    }

    return result

if \_\_name\_\_ ==  '\_\_main\_\_':

    app.run(debug=True)





1. Create a form on the frontend that, when submitted, inserts data into MongoDB Atlas. Upon successful submission, the user should be redirected to another page displaying the message **"Data submitted successfully"**. If there's an error during submission, display the error on the same page without redirection.

Ans:- Create a directory by the name “flask\_tute” after that make two folders and name them as “back-end” and “ front-end”.

For “backend”:-

1. Create a .env file and insert your MongoDB Atlas connection server URL.
2. Create a requirement.txt file and add the requirement such as

* pymongo
* dnspython
* python-dotenv

1. app.py code :-
2. from flask import Flask, request, jsonify
3. from dotenv import load\_dotenv
4. import os
5. import pymongo
6. load\_dotenv()
7. MONGO\_URI = os.getenv("MONGO\_URI")
8. client = pymongo.MongoClient(MONGO\_URI)
9. db = client.test
10. collection = db['flask\_assingment']
11. app = Flask(\_\_name\_\_)
12. @app.route('/submit', methods=['GET', 'POST'])
13. def submit():
14. form\_data = dict(request.json)
15. collection.insert\_one(form\_data)
17. return 'data submitted successfully!'
18. @app.route('/view')
19. def view():
20. data = collection.find()
21. data = list(data)  # Convert cursor to list for easier manipulation
22. for item in data:
23. print(item)
24. del item['\_id']  # Remove the MongoDB ObjectId field for cleaner output
25. data={
26. 'data': data,
27. }
28. return jsonify(data)
30. if \_\_name\_\_=='\_\_main\_\_':
31. app.run(host='0.0.0.0',port=9000,debug=True)

For front-end :-

1. Create a folder name as “templates” and make an “index.html” file in it .

Code for index.html :-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Flask\_tutedude\_assingment</title>

</head>

<body>

    <h1>Welcome to the sign up page</h1>

    <div class="centred">

        <form action="/submit" method="POST">

            <label for="username">Username:</label>

            <input type="text" name="name" placeholder="Enter your name" required>

            <br>

            <label for="email">Email:</label>

            <input type="email" name="email" placeholder="Enter your email" required>

            <br>

            <label for="password">Password:</label>

            <input type="password" name="password" placeholder="Enter your password" required>

            <br>

            <label for="confirm\_password">Confirm Password:</label>

            <input type="password" name="confirm\_password" placeholder="Confirm your password" required>

            <br>

            <input type="submit" value="Submit">

        </form>

        <a href="/get\_data"><button>View Data</button></a>

</body>

<style>

    body {

        background-color: whitesmoke;

        font-family: Arial, sans-serif;

    }

    h1 {

        text-align: center;

        color: black;

        margin-top: 20px;

    }

    .centred {

        display: flex;

        flex-direction: column;

        align-items: center;

        justify-content: center;

        height: 100vh;

    }

    form {

        display: flex;

        flex-direction: column;

        gap: 10px;

    }

    label {

        font-weight: bold;

    }

    input[type="text"],

    input[type="email"],

    input[type="password"] {

        width: 300px;

        padding: 10px;

        border-radius: 5px;

        border: 1px solid #ccc;

    }

    input[type="submit"],

    button {

        width: 150px;

        padding: 10px;

        border-radius: 5px;

        border: none;

        background-color: #007bff;

        color: white;

        cursor: pointer;

    }

    input[type="submit"]:hover,

    button:hover {

        background-color: #0056b3;

    }

    a {

        margin-top: 20px;

        text-decoration: none;

    }

    a button {

        background-color: #28a745;

    }

    a button:hover {

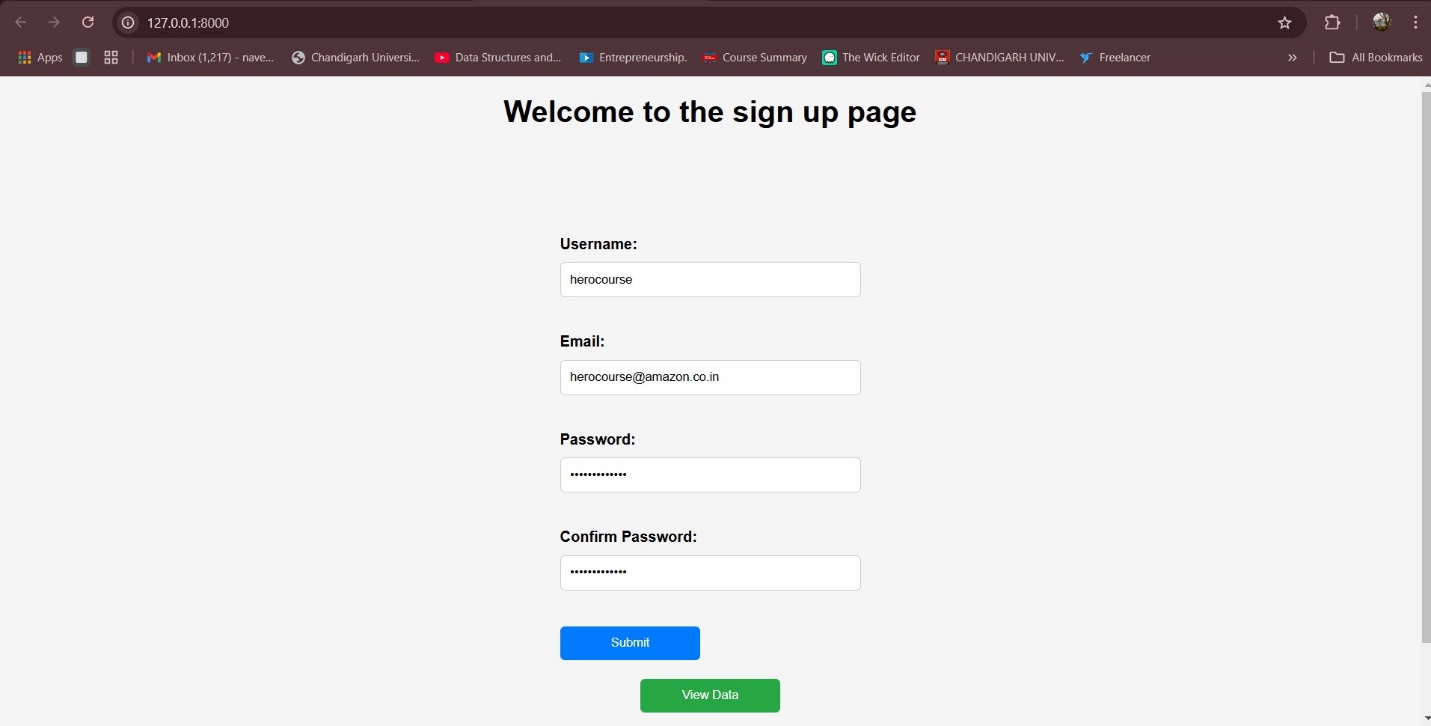
        background-color: #218838;

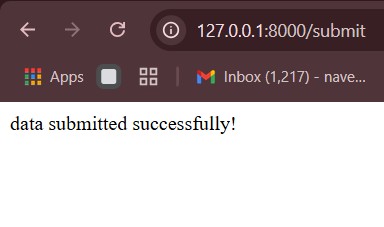
    }

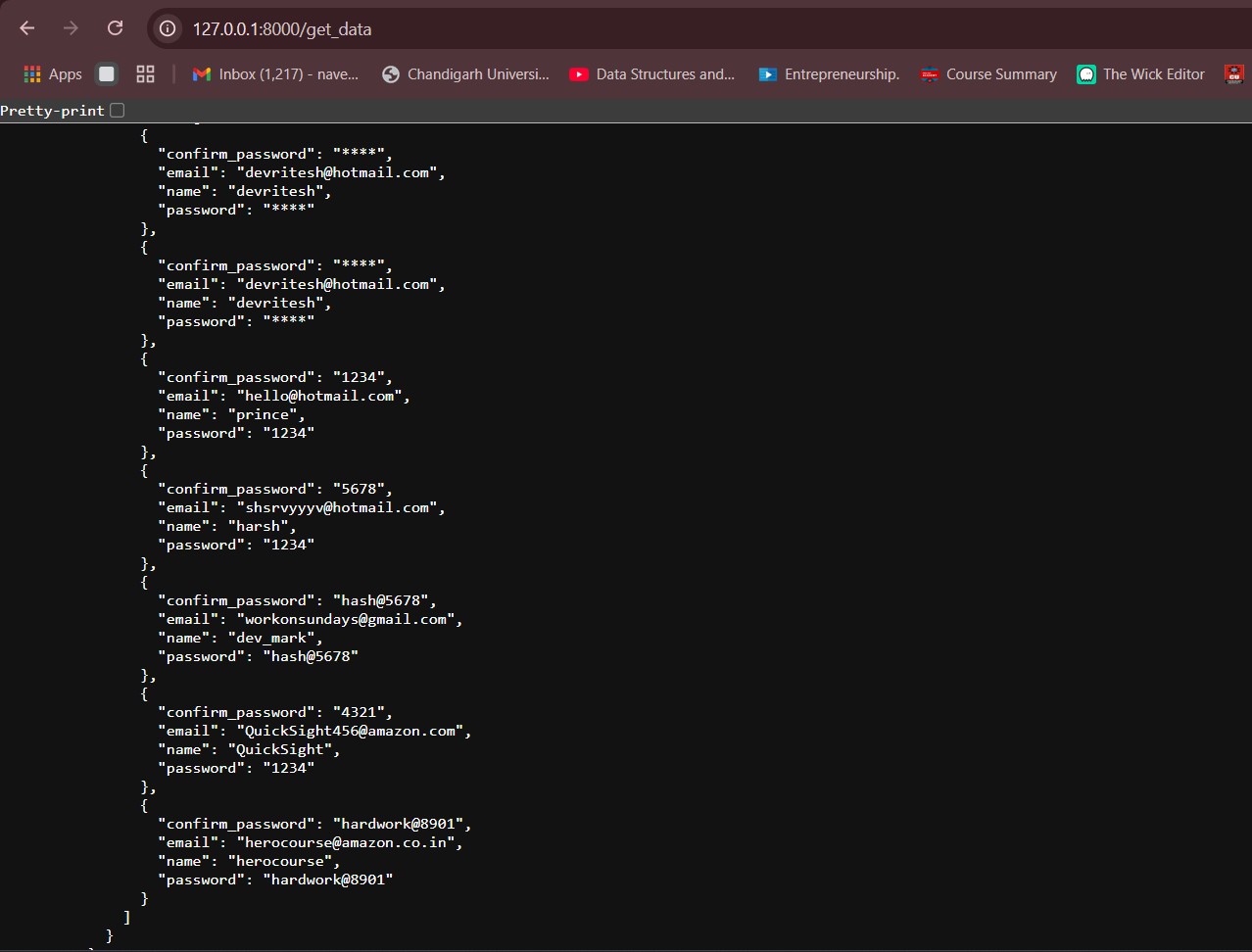
</style>

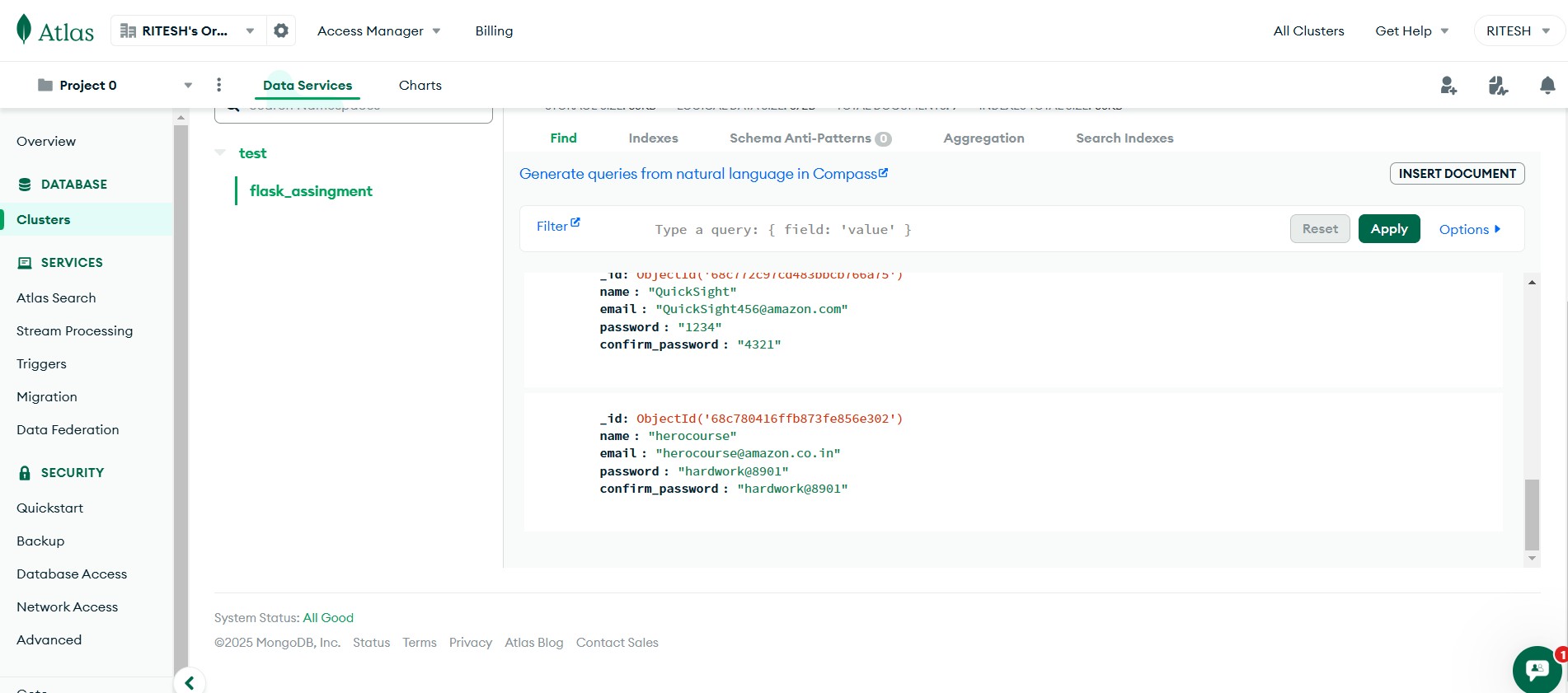
</html>

1. Now the front-end Python program for the connection with the back-end :-
2. from flask import Flask, request, render\_template
3. from datetime import datetime
4. import requests
5. BACKEND\_URL = 'http://0.0.0.0:9000'
6. app = Flask(\_\_name\_\_)
7. @app.route('/')
8. def home():
9. day\_of\_week = datetime.today() .strftime('%A')
10. current\_time = datetime.now().strftime('%H:%M:%S')
11. return render\_template('index.html', day\_of\_week = day\_of\_week, current\_time=current\_time)
12. @app.route('/submit', methods=['POST'])
13. def submit():
14. form\_data = dict(request.form)
15. requests.post(BACKEND\_URL + '/submit', json=form\_data)
16. return 'data submitted successfully!'
17. @app.route('/get\_data')
18. def get\_data():
19. response = requests.get(BACKEND\_URL + '/view')
20. return response.json()
21. if \_\_name\_\_=='\_\_main\_\_':
22. app.run(host='0.0.0.0',port=8000,debug=True)









**Submission Guidelines -:** Attach Screenshots or command along with explanation and submit in doc(google doc or microsoft doc) format also attach github repo link