SOFTWARE ENGINEER – CASE STUDY

Problem

Develop a console application to send emails using a template using one of the following programming languages: **Go, Python, Java, and Scala**.

The company X wants to develop a console application to send out marketing email to its potential customers. The marketing team has already prepared an email template for the campaign. The email template has some placeholders like **{{TITLE}}, {{FIRST_NAME}}, {{LAST_NAME}}** that need to be filled with customer information.

The email template is stored in a JSON file (**email_template.json**) which has the following format:

```
1
2  {
3      "from": "The Marketing Team<marketing@example.com",
4      "subject": "A new product is being launched soon...",
5      "mimeType": "text/plain",
6      "body": "Hi {{TITLE}} {{FIRST_NAME}} {{LAST_NAME}},\nToday, {{TODAY}}, we would like to tell you that... Sincerely,\nThe Marketing Team"
7  }
```

Notes: There is a special placeholder **{{TODAY}}** in the email template. The application should automatically replace it with the date on which it runs. The format is "31 Dec 2020"

Customers are stored in a CSV file (**customers.csv**), which has the following format:

```
1  TITLE,FIRST_NAME,LAST_NAME,EMAIL
2  Mr,John,Smith,john.smith@example.com
3  Mrs,Michelle,Smith,michelle.smith@example.com
```

The console application should do:

- Receive path to email template file and customers file as argument
- For each customer, merge customer information with the email template and save the output email into a JSON file similar to this example

```
1
2  {
3      "from": "The Marketing Team<marketing@example.com",
4      "to": "john.smith@amail.com",
5      "subject": "A new product is being launched soon...",
6      "mimeType": "text/plain",
7      "body": "Hi Mr John Smith,\nToday, 31 Dec 2020, we would like to tell you that... Sincerely,\nThe Marketing Team"
8  }
```

- Write the output email files to a folder where the path is provided as an argument to the application
- If a customer doesn't have an email address, append that customer to the **errors.csv** file (the format should be the same as the **customers.csv** file)

For example, the python application can be run as:

```
python send_email.py /path/to/email_template.json /path/to/customers.csv
/path/to/output_emails/ /path/to/errors.csv
```

Expected result

- The application must be tested with unit tests using any unit testing frameworks
- The application should be designed so that it can be updated in the future to send emails via SMTP or REST API without changing the current code base (can add new code)
- The provided solution should include a README file which describes how to build and run the application
- If possible, package the application in a Docker image so that it can run without any dependencies