

How to run and explain

1. To execute and run the code follow the below commands:

Run Golang in the terminal:

```
export $(xargs < .env)
go run *.go
```

Run dependency with docker-compose:

```
docker-compose up
```

For initial DB:

```
docker-compose -f docker-compose.yml exec liquibase /bin/sh
```

```
liquibase update --url=jdbc:postgresql://db:5432/user_db?currentSchema=public
--changelog-file=changelog.xml --username=postgres --password=postgres
```

Open Database:

```
http://127.0.0.1:8081/?pgsql=db&username=postgres&db=user_db&ns=public
```

RUN Tests:

```
TestStoreUser()
```

This is my gin framework article published in DevOps.dev [\[1\]](#). We can also add open telemetry and Prometheus to the gin middlewares.

For better documents, we can add swagger[\[2\]](#).

In this assignment, I just wrote simple integration tests but we can add unit tests, and perform tests [\[3\]](#).

For handling high transaction requests it is much better to use a Redis.

2. Solution :

1. At first change the string into the map of characters and count each element
 2. Then choose max counter and select the character as the first element
 3. Then choose another max character
 4. Repeat steps 2 and 3 until finish characters.
3. Run the main
4. Run the tests