Test 8

The problems below are divided into three levels - A, B, and C - with increasing level of challenge. Higher consideration will be given to those who can solve all three problems.Make any assumptions required to fill in potential gaps in completion of the problem statement requirements. However, clearly explain your assumptions and comment your code well.

Level A

Create a program in Golang to read a YAML file and store it into a simple Bolt Database. BoltDB is a simple database built for Golang with key-value based storage. Link: https://github.com/boltdb/bolt.

Specifications

- Read the attached YAML file into a Go Structure.
- Use this Structure to create a Bolt DB entry
- Buckets should be Jobs, Skills and Candidates
- Key-Value Pairs should be second level items in YAML. E.g(Key - Role1 Value - FullStack)
- Create and fill Bucket Values from the structure created by reading YAML file. Do not hard-code these values.

Create a BoltDB key value store to read the attached YAML file (test.yaml) into a structure and then save it into DB:

```yaml
---
Jobs:
  Role1: FullStack
  Role2: AiDeveloper
Skills:
  FullStack: Frontend, Backend
  AiDeveloper: MachineLearning, Backend
Candidates:
 Ace: Role1
 Bose: Role2
 Crank: Role1
 Dole: Role2
 Easter: Role2
```

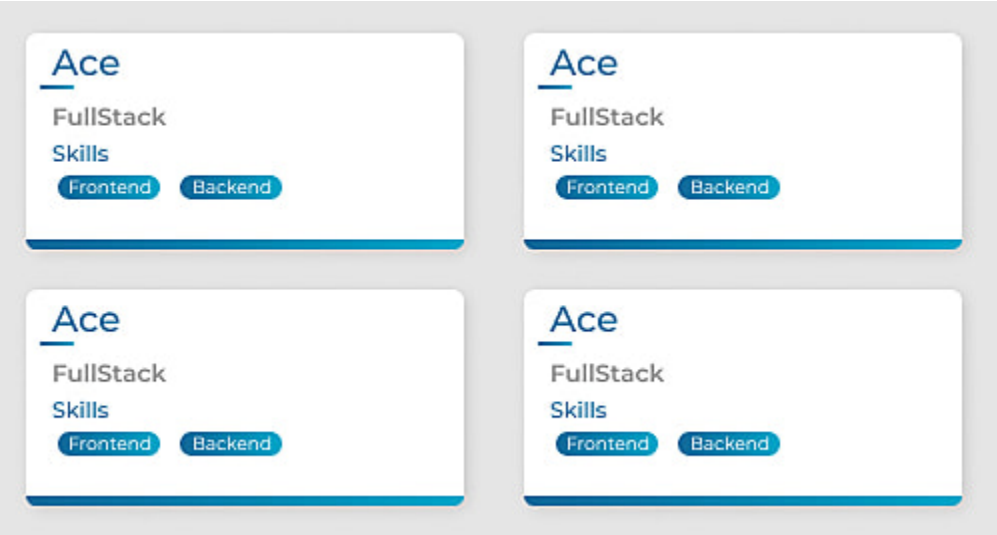
```
14  Folk: Role1
```

Level B

Create a Server in Golang and UI(Bootstrap/CSS) to display the data from DB in the form of Cards.

Specifications

- Each card should display a Candidate's name, their Job, and their Skills. E.g. Ace, FullStack, MachineLearning, Backend will be displayed on Ace's Card. (Note that in Candidates Bucket, keys are Candidate Names). See example below -
- Show 6 cards for all candidates on the same page.



Level C

Create a responsive web page that can switch between desktop and mobile versions on change of window size to smaller window. Cards should be stacked vertically in the mobile/small-window view.