Report 8

Team members:
Dias Galikhanov 200103092
Yermukhamed Tilek 200103094
Baijuman Ali 200103514
Dauirbek Sanakuly 200103453
Zhumadil Kenzhebay 200103245

For this week we have done:
Wrote Dockerfile for our project with multi-stage building.

We wrote Dockerfile which will create docker image of our project. It is well optimized and won't use much memory. For this we used multi-stage building.

```
ockerfile > ...
  FROM golang:latest AS builder
  WORKDIR /app
  COPY go.mod .
  COPY go.sum .
  RUN go mod download
  COPY . .
  RUN GO111MODULE="on" CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o app ./cmd

  FROM alpine:latest
  WORKDIR /app
  COPY --from=builder /app/app .
  COPY --from=builder /app/config /app/config
  COPY --from=builder /usr/local/go/lib/time/zoneinfo.zip /
  ENV TZ=Asia/Almaty
  ENV ZONEINFO=/zoneinfo.zip
  CMD ["./app"]
```

Here you can see we divided process into two stages. First is "builder". Here we copy all our files and set all dependencies and compile our project and create that compiled file. In second stage we used alpine image which is one of the images with the least amount of memory. It is just virtual env which don't have anything even timezone. We copy our compiled binary file to our alpine image and set some configs and timezone. It works and we saved a lot of memory.

But it won't work on our project now. Because we have to connect our app with database and in our docker container there is no database. For this people use docker-compose which runs multiple containers. We've created docker-compose.yml file but it is empty. So far we don't know how to use it and are studying it.

For the next week:

We plan to finish our job with docker-compose. And make our project able to run in docker containers.