

Deployment Guide

Deployment Guide:

Welcome to the BoxJelly project. As a user who wants to deploy and use this project, it is important to understand that the project consists of two parts: the front-end web page, which displays data charts, and the back-end, which communicates with the Garmin Connect platform and processes data. The back-end also includes a basic database program used for storing data. To properly deploy this project and make it fully functional, the front-end, back-end, and database must all be completed in sequence according to the steps in this guide.

1. DataBase:

required Debian 10 instance (Recommended service provider: Vultr, AWS)

1. Import MongoDB's public GPG key:

First, import MongoDB's public GPG key to ensure the integrity and authenticity of the downloaded packages. Run the following command:

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

2. Add the MongoDB repository:

Create a list file containing the MongoDB repository. For MongoDB 4.4, run the following command:

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/debian buster/mongodb-org/4.4 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```

Then, update the local package database:

```
sudo apt update
```

3. Install MongoDB using the following command:

```
sudo apt install -y mongodb-org
```

4. Start the MongoDB service:

After the installation is complete, start the MongoDB service and set it to start automatically at system boot:

```
sudo systemctl start mongod
sudo systemctl enable mongod
```

5. Verify the MongoDB installation:

Check the status of the MongoDB service using the following command:

```
sudo systemctl status mongod
```

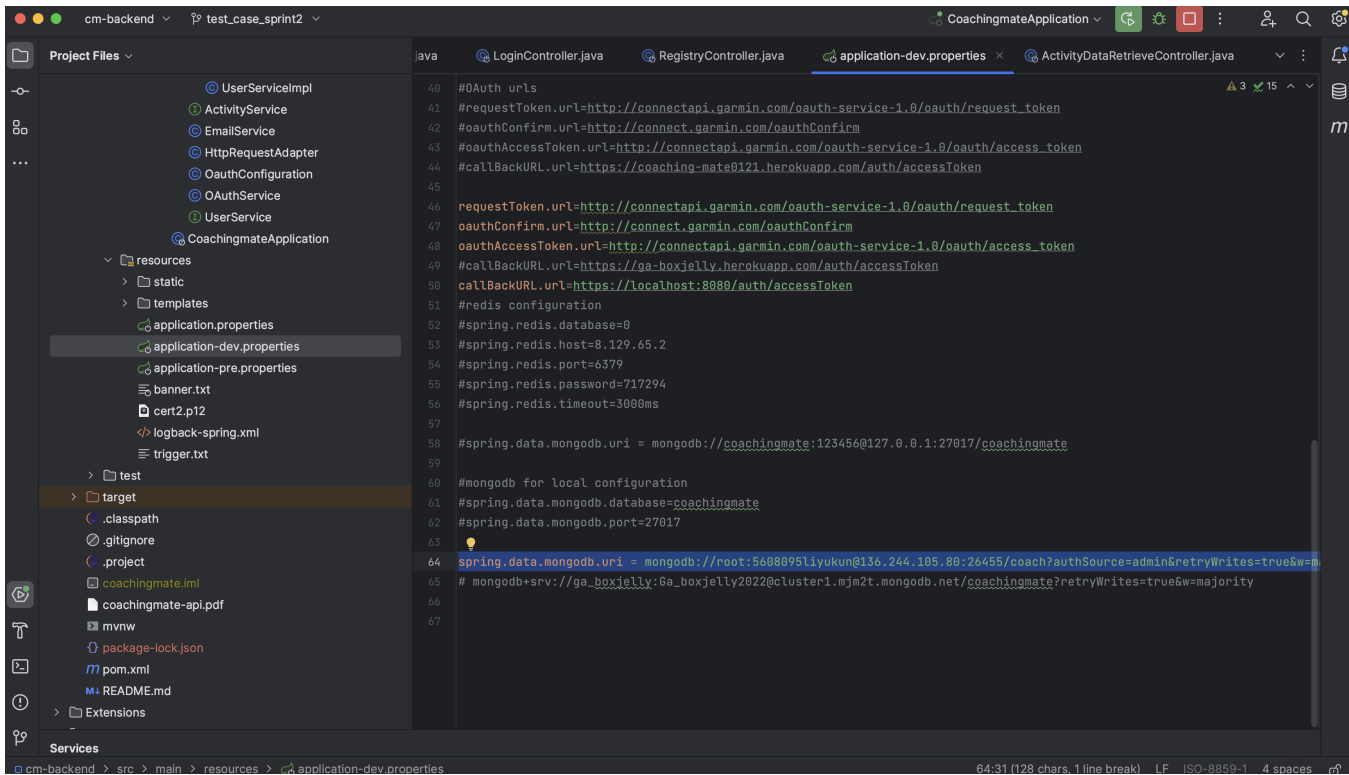
If you see Active: active (running) status, it means MongoDB has been successfully installed and is running.

Now, you have successfully deployed MongoDB on a Vultr Debian 10 instance. To ensure system security, make sure to properly configure MongoDB, manage permissions, and set up the firewall when deploying it in a real-world scenario. For more information, refer to the MongoDB official documentation (<https://docs.mongodb.com/manual/administration/security-checklist/>).

2. BackEnd:

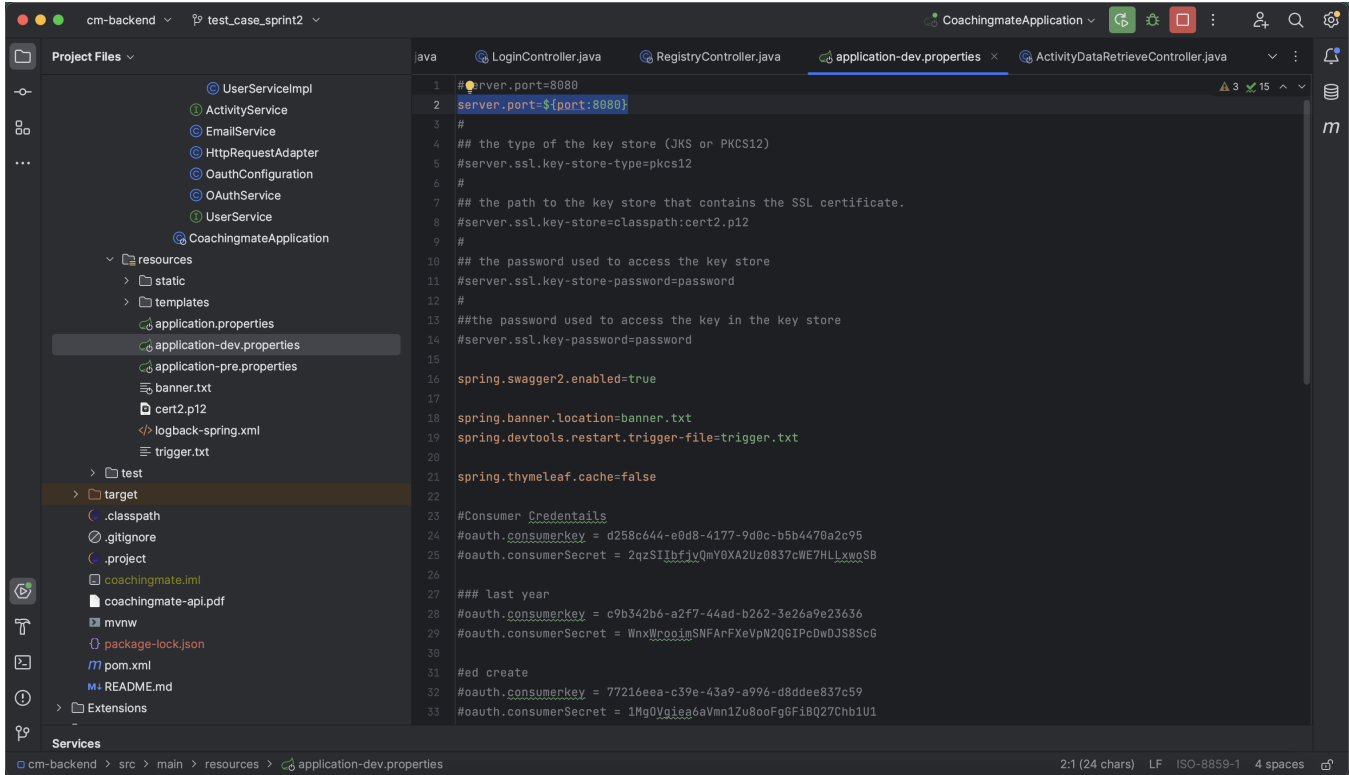
Change the database of your own (ignore this if you want to use the old database from the previous year)

Note: The old database already has a lot of sample data inside. it can be harder to find the new data added



```
40 #OAuth urls
41 #requestToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/request_token
42 #oauthConfirm.url=http://connect.garmin.com/oauthConfirm
43 #oauthAccessToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/access_token
44 #callbackURL.url=https://coaching-mate0121.herokuapp.com/auth/accessToken
45
46 requestToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/request_token
47 oauthConfirm.url=http://connect.garmin.com/oauthConfirm
48 oauthAccessToken.url=http://connectapi.garmin.com/oauth-service-1.0/oauth/access_token
49 #callbackURL.url=https://ga-boxjelly.herokuapp.com/auth/accessToken
50 callbackURL.url=https://localhost:8080/auth/accessToken
51
52 #redis configuration
53 #spring.redis.database=0
54 #spring.redis.host=8.129.65.2
55 #spring.redis.port=6379
56 #spring.redis.password=717294
57 #spring.redis.timeout=3000ms
58
59 #spring.data.mongodb.uri = mongodb://coachingmate:123456@127.0.0.1:27017/coachingmate
60
61 #mongodb for local configuration
62 #spring.data.mongodb.database=coachingmate
63 #spring.data.mongodb.port=27017
64
65 spring.data.mongodb.uri = mongodb://root:5688895liyuKun@136.244.105.80:26455/coach?authSource=admin&retryWrites=true&w=
66 # mongodb+srv://ga_boxjelly:Ga_boxjelly2022@cluster1.mjm2t.mongodb.net/coachingmate?retryWrites=true&w=majority
67
```

If the default port 8080 is occupied locally, it needs to be modified.



```
1 #server.port=8080
2 server.port=${port:8080}
3 #
4 ## the type of the key store (JKS or PKCS12)
5 #server.ssl.key-store-type=pkcs12
6 #
7 ## the path to the key store that contains the SSL certificate.
8 #server.ssl.key-store=classpath:cert2.p12
9 #
10 ## the password used to access the key store
11 #server.ssl.key-store-password=password
12 #
13 ##the password used to access the key in the key store
14 #server.ssl.key-password=password
15
16 spring.swagger2.enabled=true
17
18 spring.banner.location=banner.txt
19 spring.devtools.restart.trigger-file=trigger.txt
20
21 spring.thymeleaf.cache=false
22
23 #Consumer Credentials
24 #oauth.consumerkey = d258c644-e0d8-4177-9d0c-b5b4470a2c95
25 #oauth.consumerSecret = 2qzSIbfjyQmY0XA2Uz0837cWE7HLxwoSB
26
27 ### last year
28 #oauth.consumerkey = c9b342b6-a2f7-44ad-b262-3e26a9e23636
29 #oauth.consumerSecret = WnxWcooimSNFARFeVpN2Q61PcDwDJS8ScG
30
31 #ed create
32 #oauth.consumerkey = 77216eea-c39e-43a9-a996-d8ddee837c59
33 #oauth.consumerSecret = 1Mg0Vg1ea6aVm1Zu8ooFg6F1BQ27Chb1U1
```

Change the callback URL of your own

1. Update the system:

First, open a terminal on your instance and run the following commands to update all packages:

```
sudo apt update
sudo apt upgrade -y
```

2. To run the jar file, you need Java installed on your instance. It is recommended to use OpenJDK. To install Java 8, run the following command:

```
sudo apt install openjdk-11-jdk -y
```

After the installation is complete, verify that Java was successfully installed:

```
java -version
```

3. Download the jar file:

First, make sure 'wget' tools are installed:

```
sudo apt install wget -y
```

Then, use one of the following commands to download the jar file:

```
wget https://github.com/COMP90082-2023-SM1/GA-BoxJelly/raw/server-YK/src/cm-backend/target/coachingmate-0.0.1-SNAPSHOT.jar
```

4. Run the jar file: Now you can run the downloaded jar file. Use the following command:

```
java -jar coachingmate-0.0.1-SNAPSHOT.jar
```

If you want to run the jar file in the background and keep it running after closing the terminal, you can use the `nohup` command:

```
nohup java -jar coachingmate-0.0.1-SNAPSHOT.jar &
```

3. FrontEnd:

Run locally

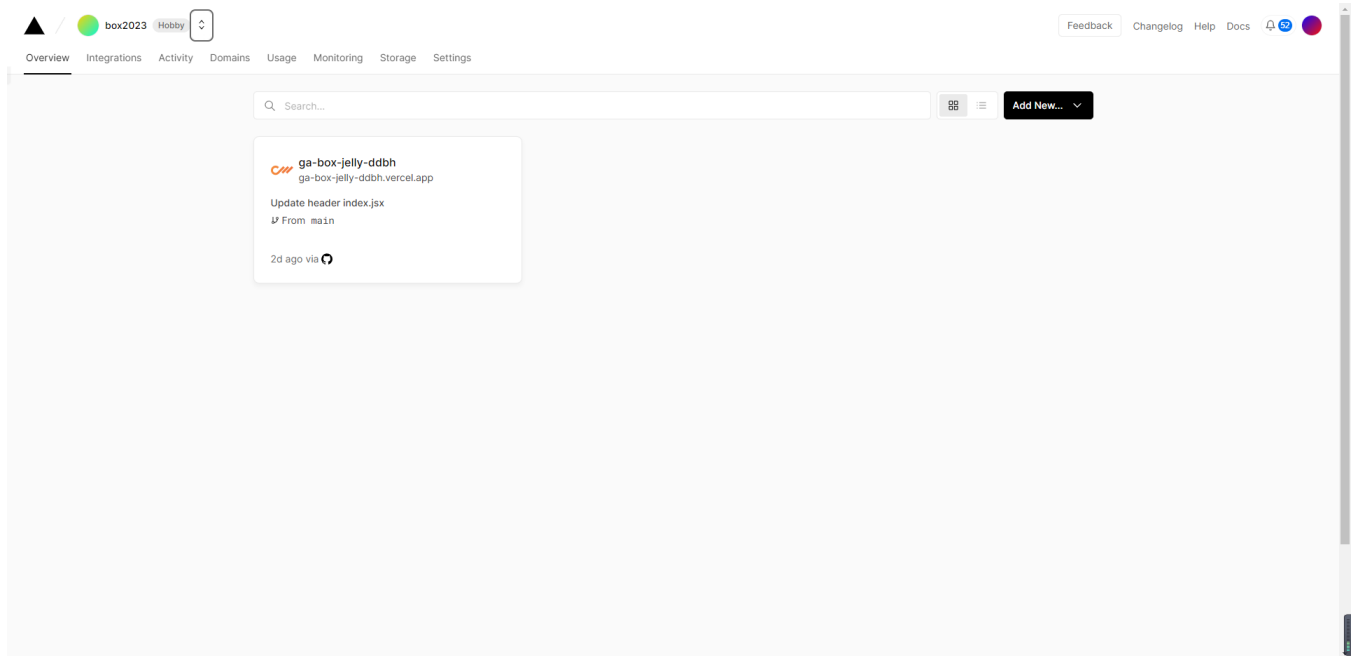
required: pre-installed: Nodejs V16.19.1

1. Navigate to src\cm-frontend-react folder
2. Run npm install
3. Run npm start

Online

Deploy online

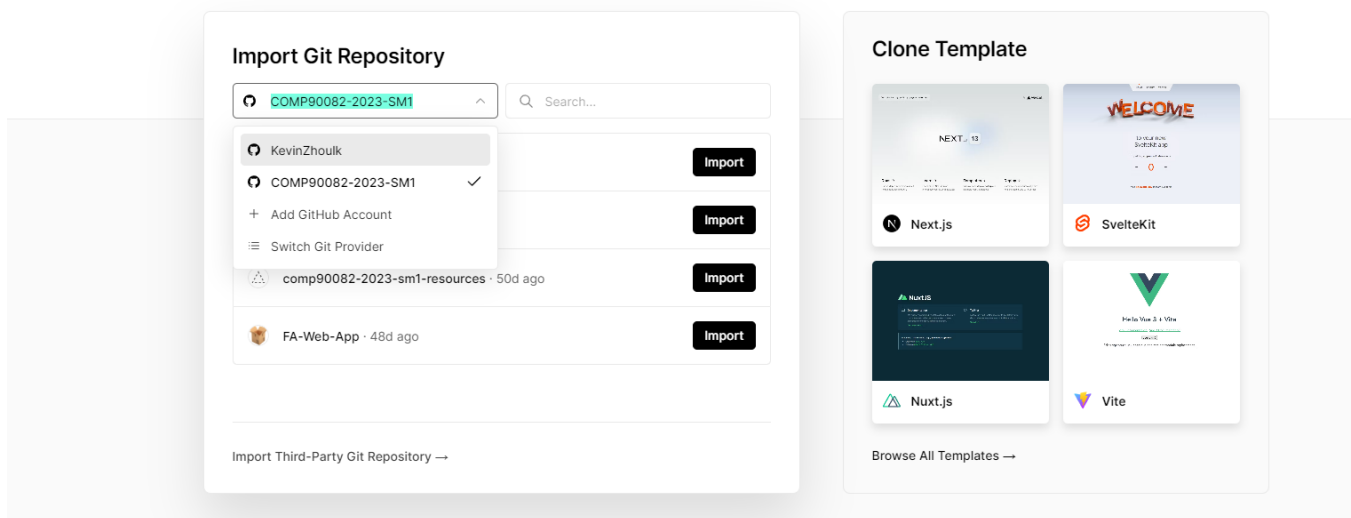
1. Create a new project on Vercel



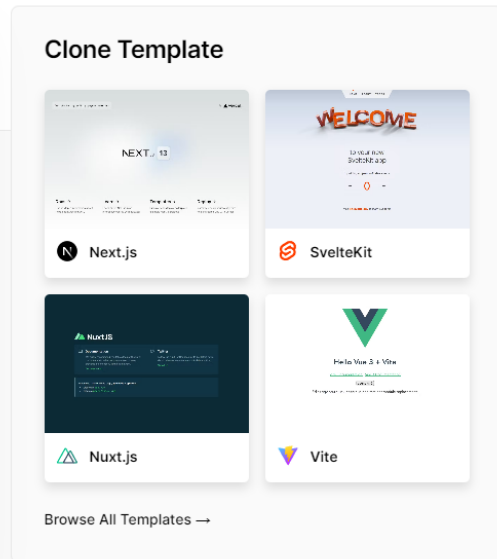
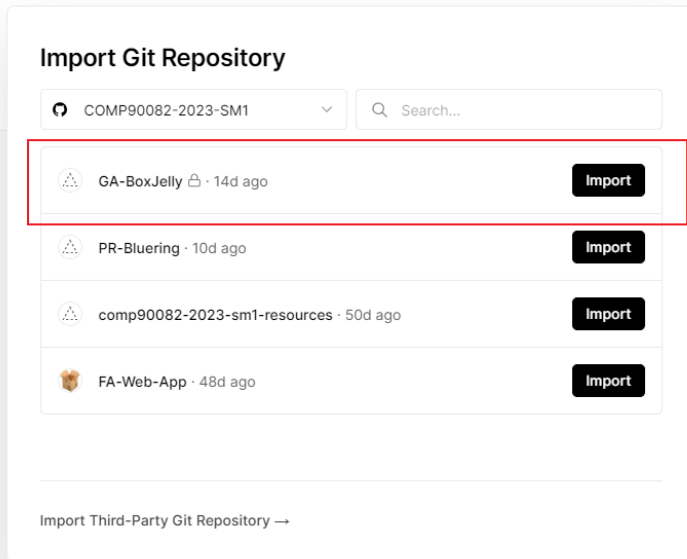
2. Connect it with Github

Let's build something new.

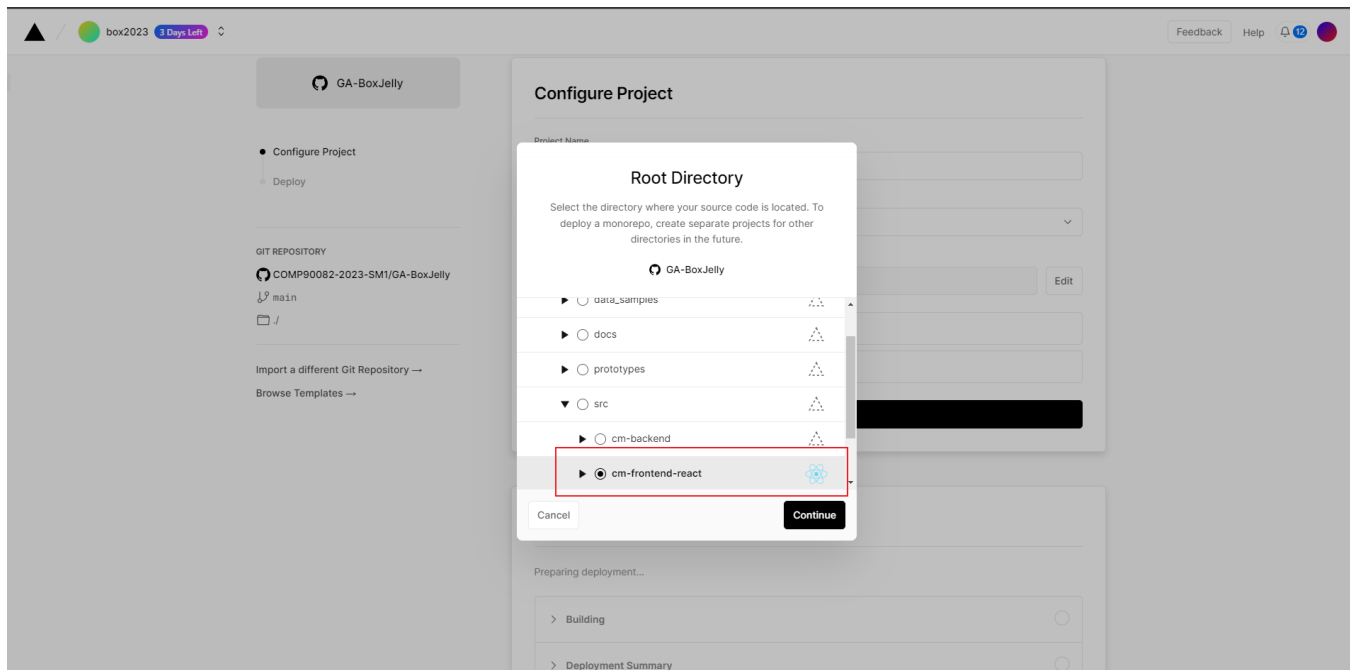
To deploy a new Project, import an existing Git Repository or get started with one of our Templates.



3. Import the required repository



4. Select the source code folder



Note:

Change the build command to npm run build

Change the install Command to npm install

The screenshot shows a deployment configuration form with the following sections:

- Project Name:** A text input field containing "ga-box-jelly-zub9".
- Framework Preset:** A dropdown menu showing "Create React App" with a gear icon and a downward arrow.
- Root Directory:** A text input field containing "src/cm-frontend-react" and an "Edit" button.
- Build and Output Settings:** A section with a downward arrow and the text "Build and Output Settings". It contains:
 - Build Command:** A text input field containing "npm run build" and an "OVERVERRIDE" toggle switch that is turned on (blue).
 - Output Directory:** A text input field containing "build" and an "OVERVERRIDE" toggle switch that is turned off (grey).
- Install Command:** A text input field containing "npm install" and an "OVERVERRIDE" toggle switch that is turned on (blue).
- Environment Variables:** A section with a right-pointing arrow and the text "Environment Variables".
- Deploy:** A large black button with the text "Deploy" in white.

Red rectangular boxes highlight the "Build Command" and "Install Command" sections.

5. Click the Deploy button.

3. Remote Backend Deployment (in case you want to deploy the product online):

1. Configure `src/main/resources/application-dev.properties`
 - a. add this line to make your backend access remotely: `server.address=0.0.0.0`
 - b. change `callBackURL.url`
 - c. configure `oauth.consumerkey` and `oauth.consumerSecret` (and set up endpoint urls in the garmin api tool)
 - d. change your database url
 - e. add your own smtp service
2. In `src/main/java/coachingmateanalytics/coachingmate/service/SendEmail.java`
 - a. Modify url in `sendResetEmail`
3. In `src/cm-frontend-react/src/utlis/request.js`
 - a. Change `baseUrl` in line 13 to your backend url