Weather Dashboard Project

Executive summary

The Weather Dashboard Project allows any user to find out the current weather conditions (degrees in Celsius and cloud cover) of the city they live in or any other city in the world. The project uses various services and applications such as Google Cloud Platform (GCP), serverless environment service Google Cloud Run, Docker for containerization and weather APIs related to the project. Using GCP provides high availability and scalability, resulting in convenient service to the user base.

Introduction

Google Cloud Platform is a cloud computing platform that provides a wide range of services like computing, machine learning, managing applications, deploying and etc. provided by Google. It's relevance in deploying scalable web applications is very huge, as it provides services like serverless environment, data storage, containerization and etc. The motivations behind choosing Google Cloud Platform for the Weather Dashboard project is that this project needs scalability, reduced operational costs, certain API's and serverless environment and containerization. Since this infrastructure provides everything needed, I chose to use GCP.

Project Objectives

- Creating Weather Dashboard Web application using React, JavaScript, CSS and HTML.
- Using third party API's like OpenWeatherMap to provide current weather conditions.
- Using Docker for containerization for easier deployment and management.
- Implementing a serverless architecture with Google Cloud Run for seamless deployment and scalability.

Google Cloud Platform Overview

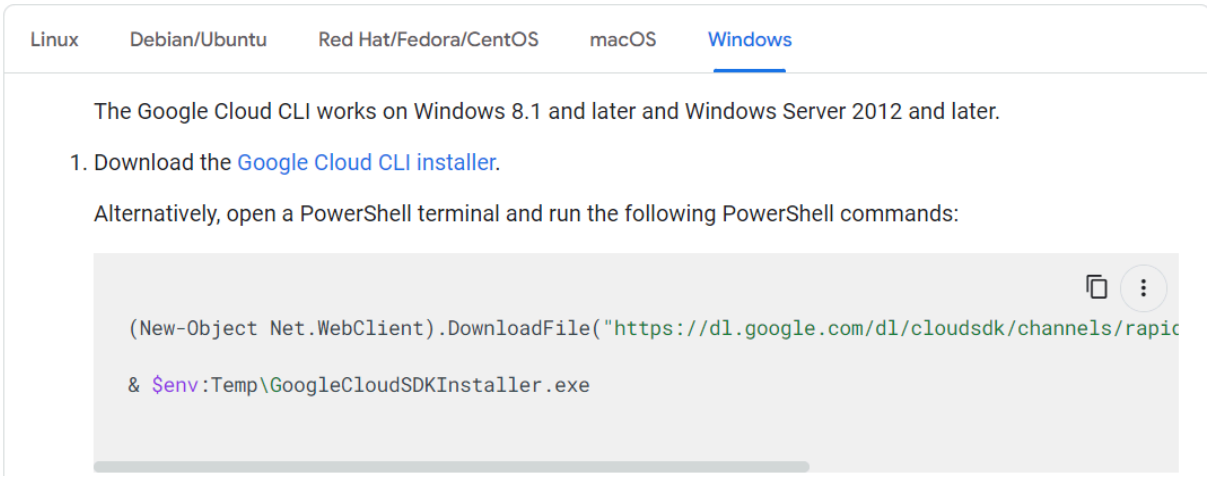
Google Cloud Platform (GCP) is a suite of cloud services that allow developers to build, deploy, and scale applications in a cloud environment. Examples for services:

- Cloud Functions
- App Engine
- Cloud Run
- Google Kubernetes Engine
- Cloud SQL
- Cloud Storage
- Cloud Firestore

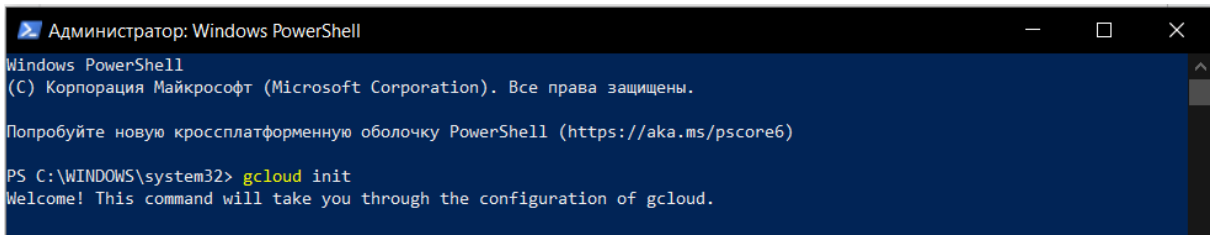
Benefits of using Google Cloud Platform are:

- Performance
- Scalability
- Security
- Availability(In terms of pricing)
- Seamless Integration
- Global Infrastructure

Google Cloud SDK and Cloud Shell



Go to the Google Cloud official website and download Google Cloud CLI installer and download contents via installer.



After downloading everything, write “gcloud init” in your terminal then authenticate your account.

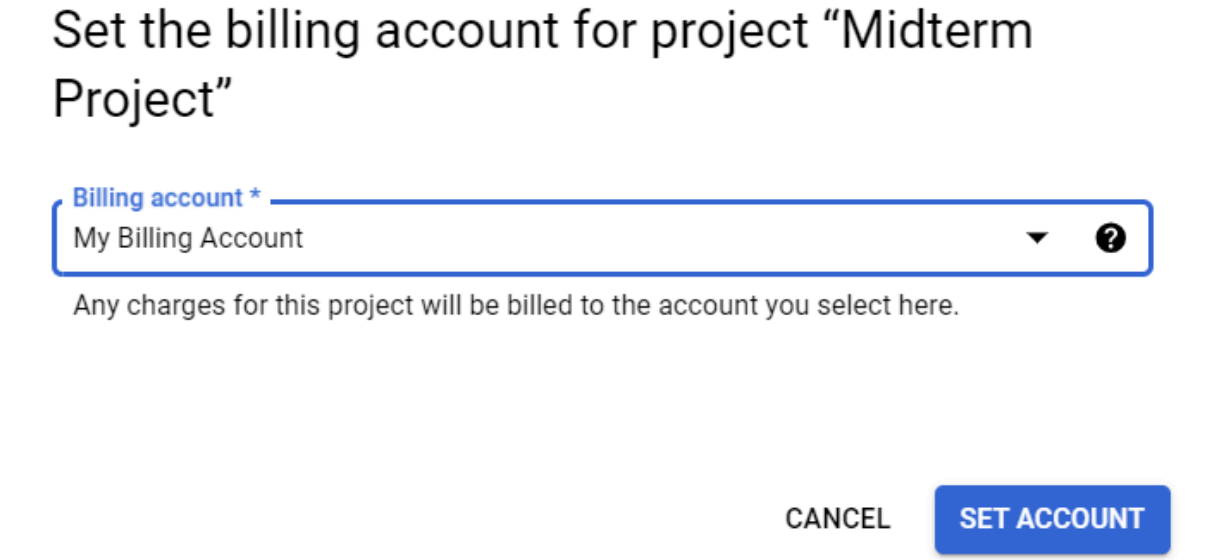
Cloud Shell was created for deploying applications and configuring resources, which allows access to the terminal without needing local configuration. It provides a convenient environment to run

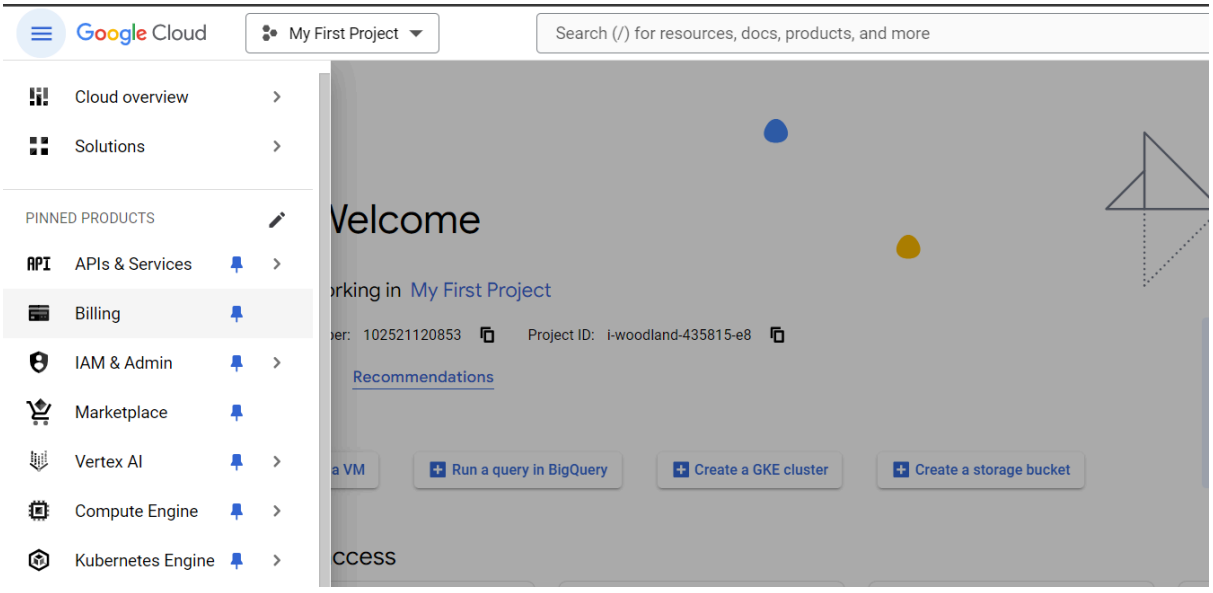
Creating Project

```
C:\Users\aliha\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_
uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fww
w.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.google
apis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Faut
h%2Faccounts.reauth&state=3NxsZaS3tJFd6sMONJgvbirQIl8AK9&access_type=offline&code_challenge=fpTh67qCIA6dQfvg_cDGhp31pjBoqE2
c3YVfyeZEXlo&code_challenge_method=S256


You are now logged in as [alihanmurat11@gmail.com].
```

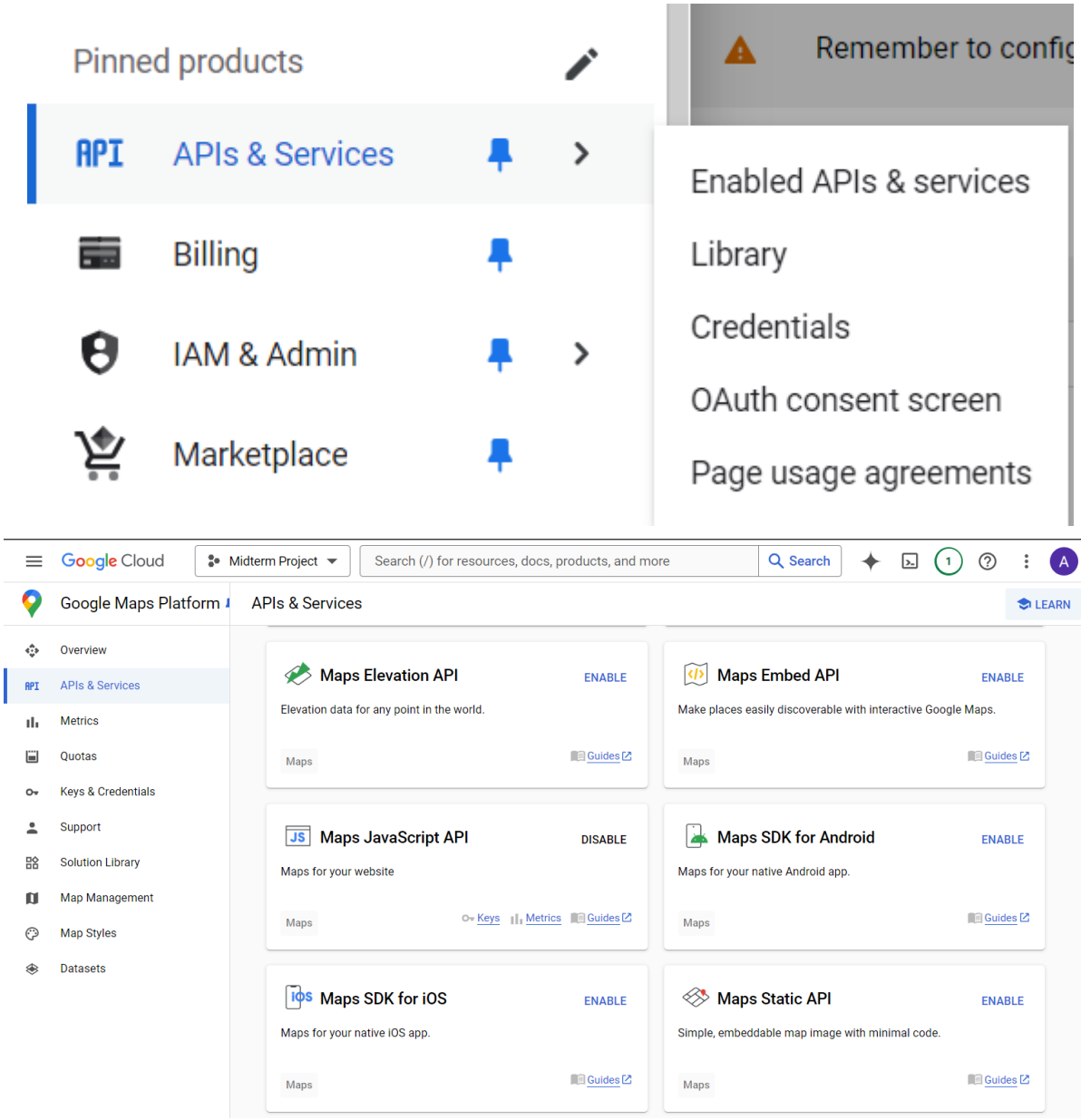
```
C:\Users\aliha\AppData\Local\Google\Cloud SDK>gcloud projects create cloud-app-dev-midterm --name="Midterm Project"
Create in progress for [https://cloudresourcemanager.googleapis.com/v1/projects/cloud-app-dev-midterm].
Waiting for [operations/cp.7840514447646538079] to finish...done.
Enabling service [cloudapis.googleapis.com] on project [cloud-app-dev-midterm]...
Operation "operations/acat.p2-425509735138-bbb1a3e0-69bb-45ab-b083-643f40fed43f" finished successfully.
```

Login to your account and create a google project, then set the to project to default project.

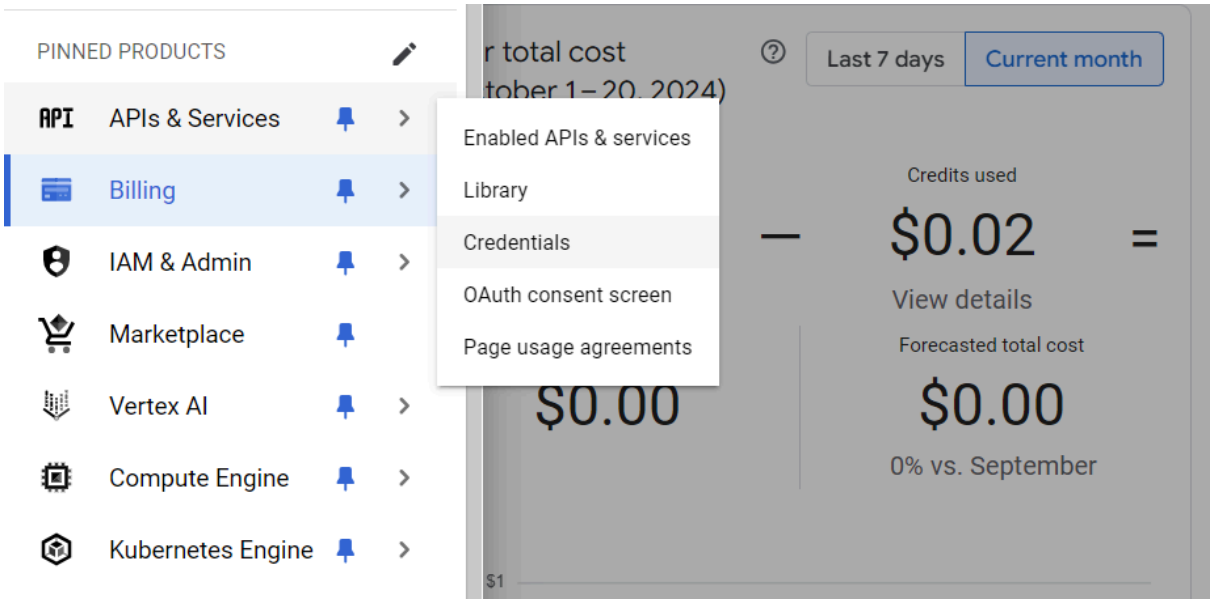


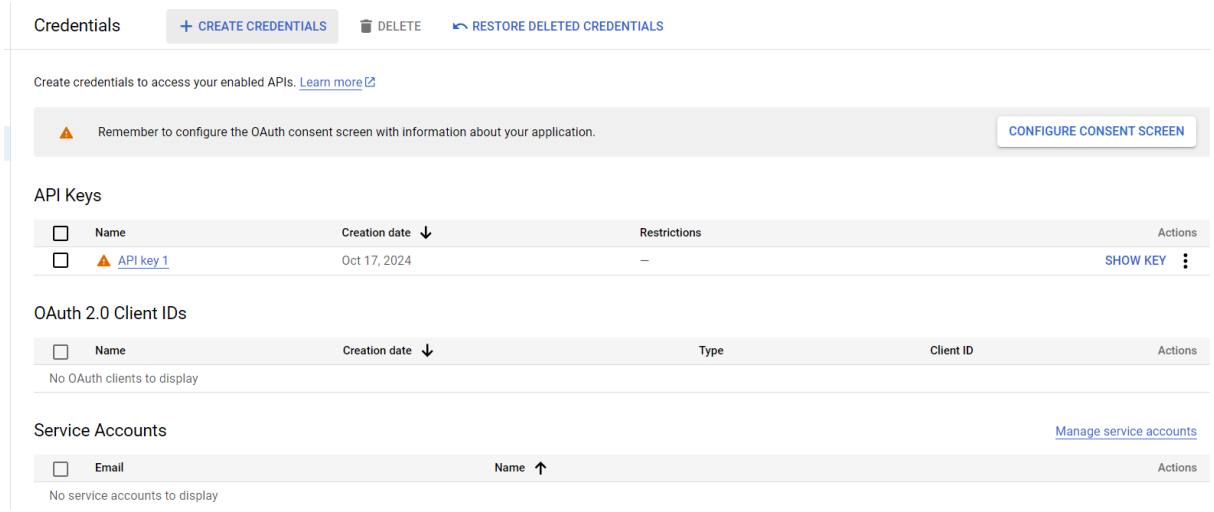


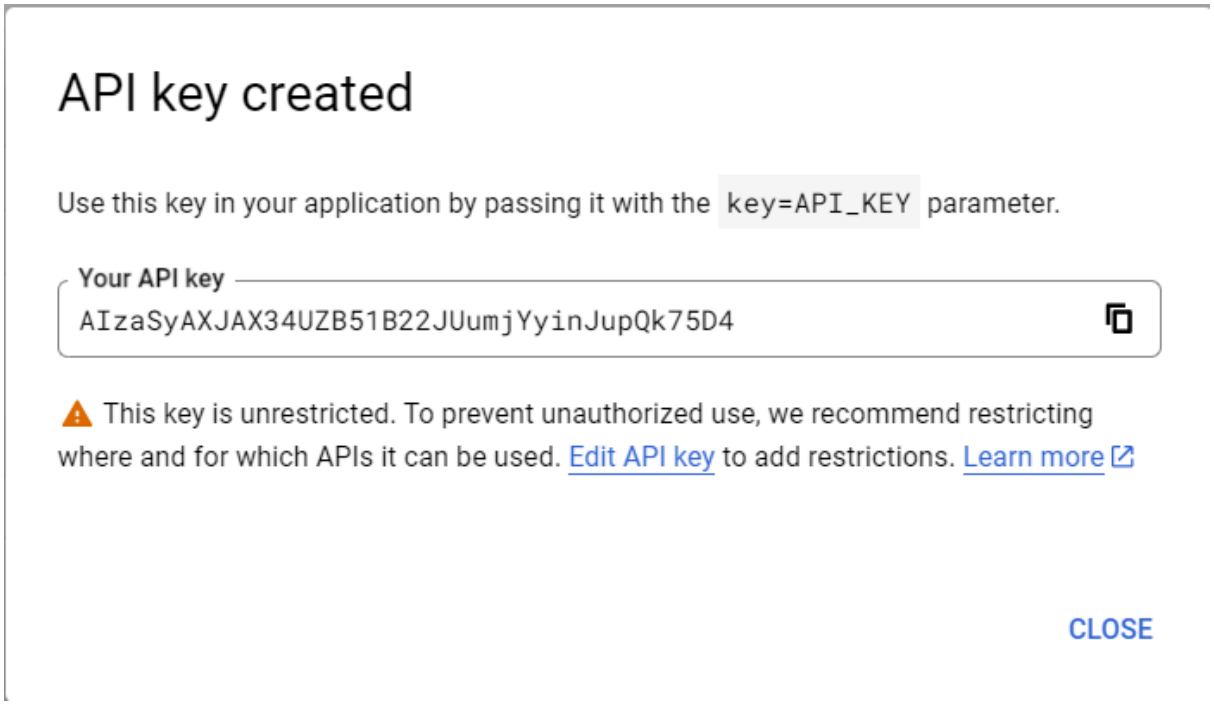
Make sure that Billing account is activated for your project, otherwise your project won’t work



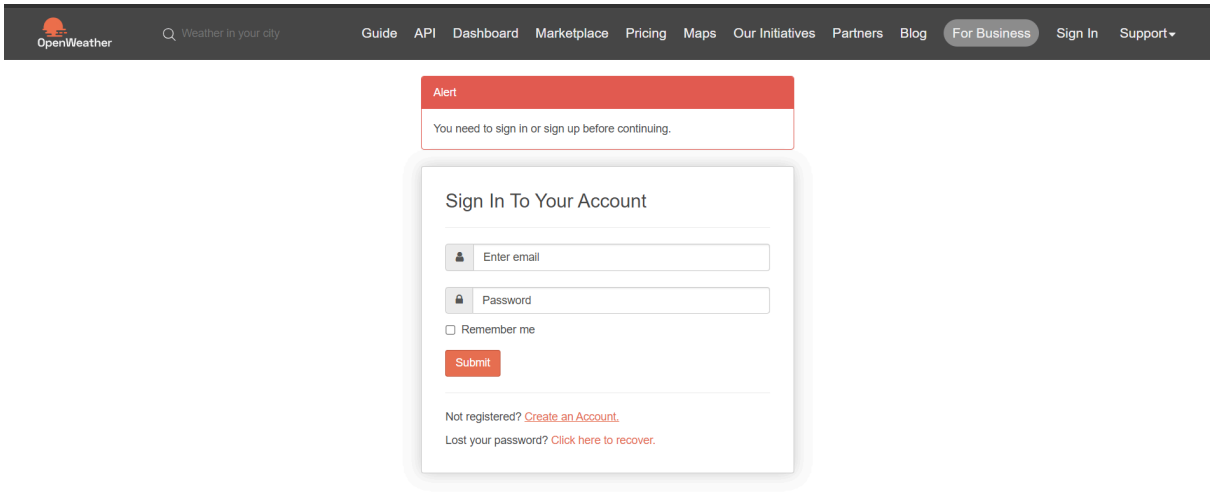
Go to the "**APIs & Services**" page, Click to the "**Enabled APIs & services**". Here you should activate APIs and services needed to your project. In my case I activated **Google Cloud Run API**, **Google Cloud Functions API**, **Cloud Storage API**, **Google Maps JavaScript API**, **OpenWeather API**.



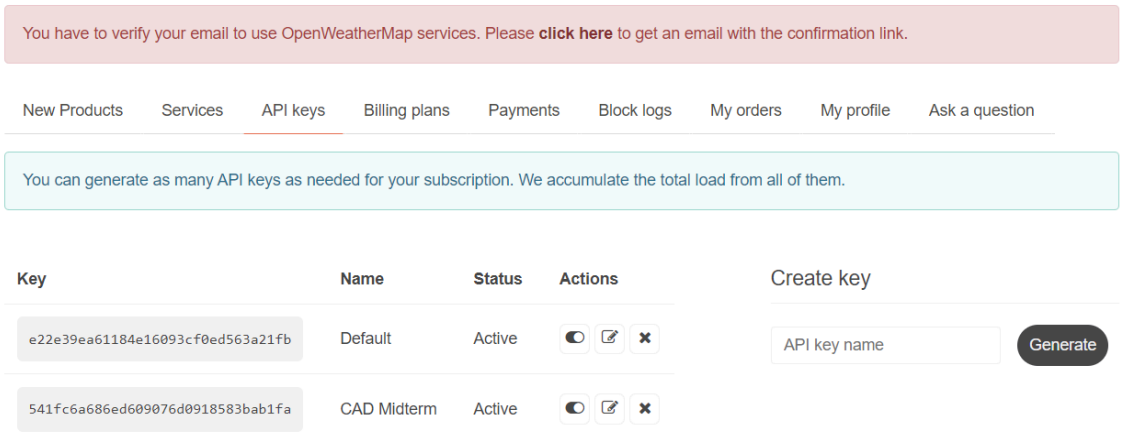




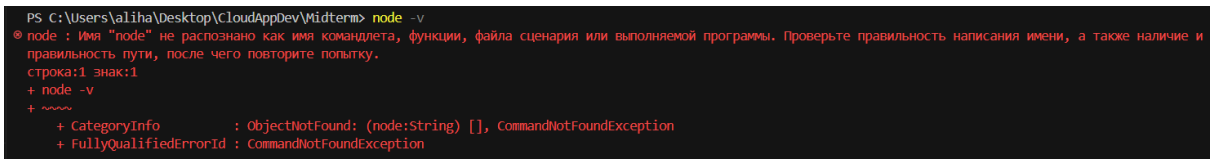
Go to the "**APIs & Services**", click to the "**Credentials**". Then click "**Create Credentials**" and choose API key. You can add this API key to your project's configuration.



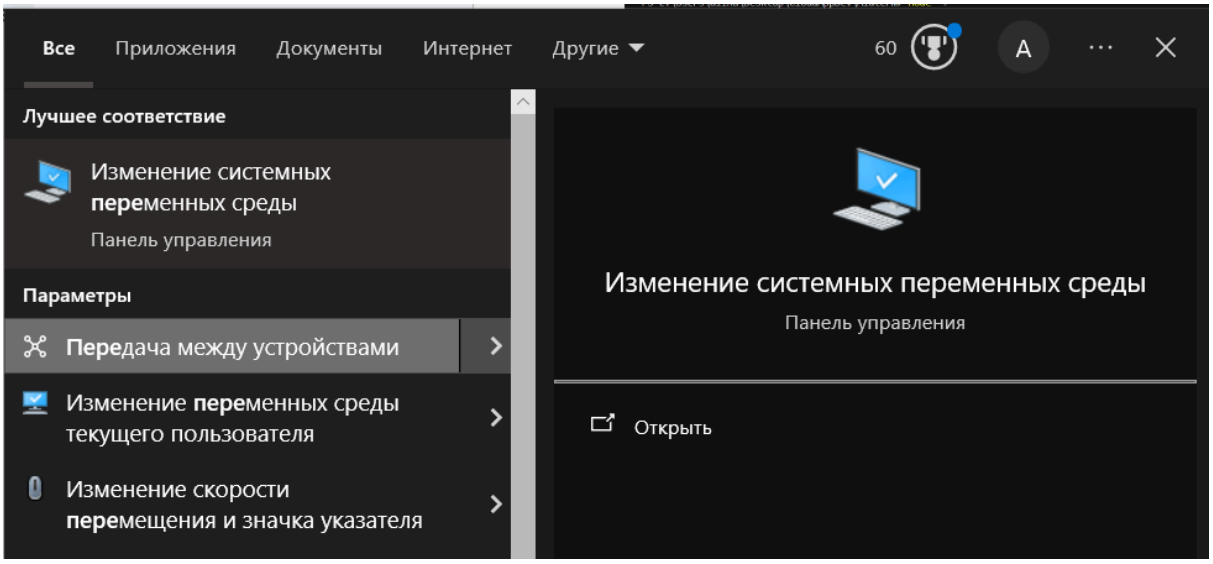
Go to the [OpenWeatherMap website] and create an account if you don't have one.



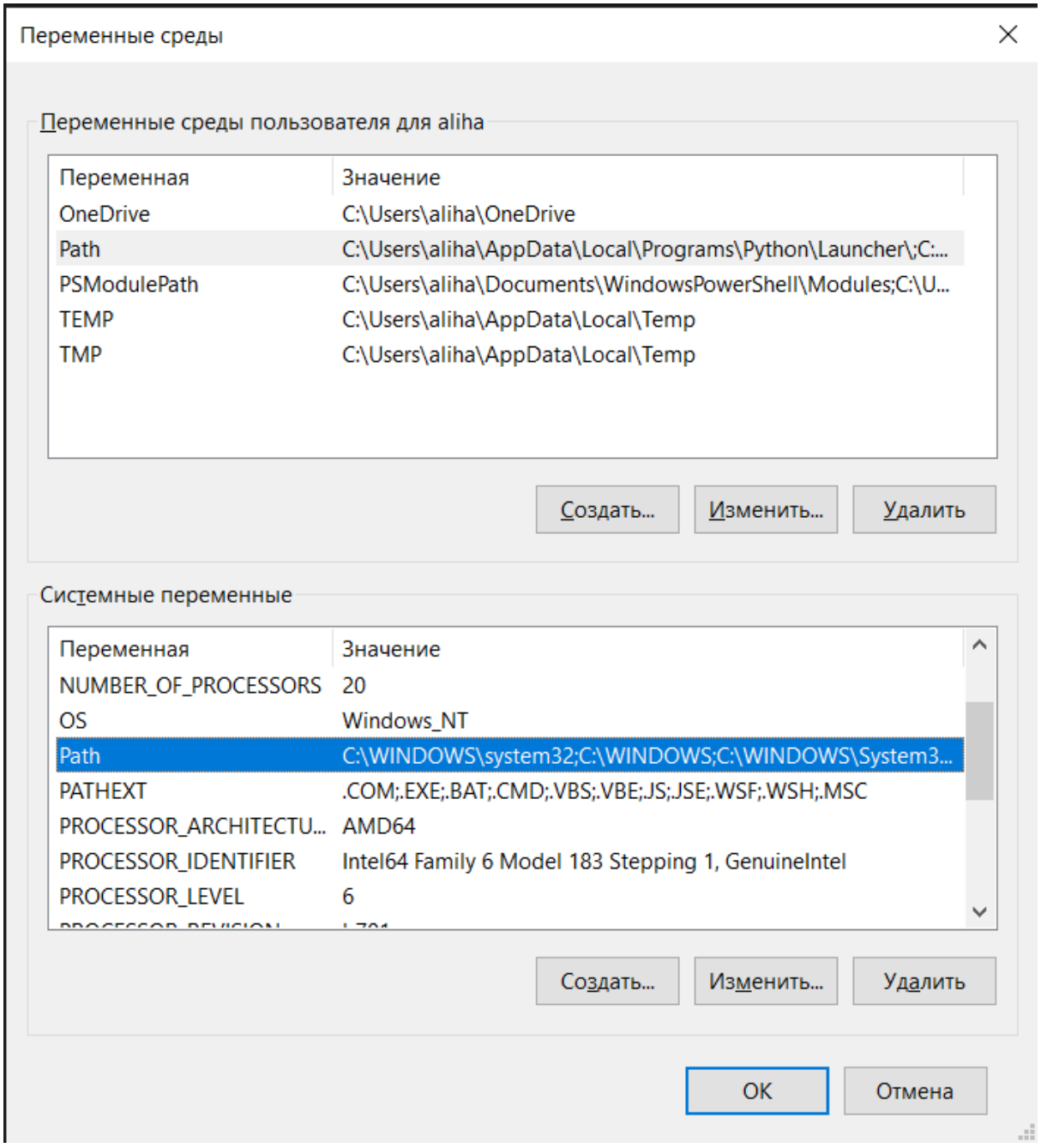
On your **Profile** click on the "**My API keys**" and generate an API key. Then copy your API key. We will use this API key on our project to get weather condition information.



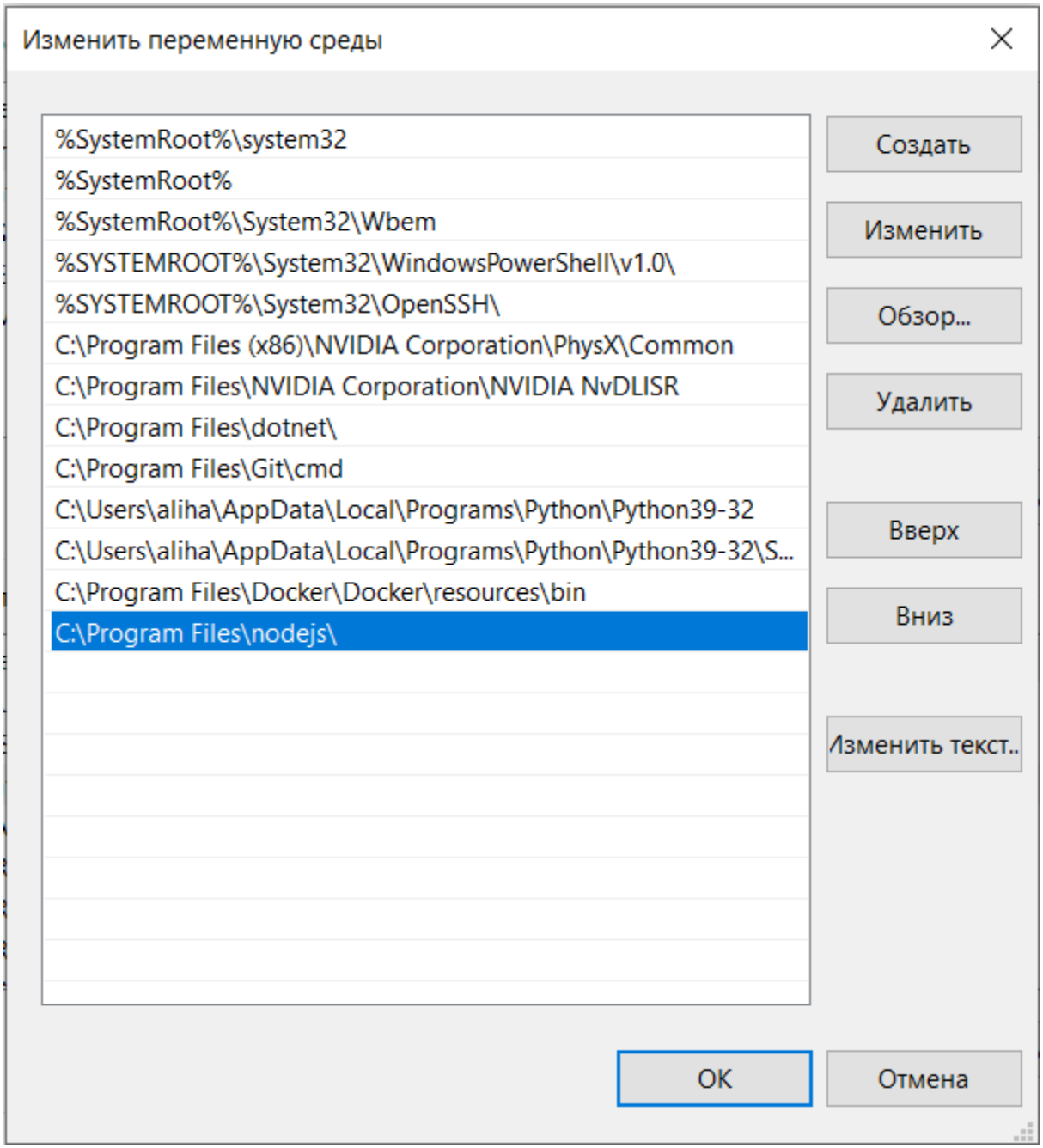
On your terminal check Node.js version. If you struggle with this type of message. First of all, you should install Node.js. Go the [Node.js official website] and install Node.js



Then search for "**Environment Variables" on your system**



Select "Path" and click "Edit"



Click “Create” and paste the directory where you installed Node.js, then save everything.

```
PS C:\Users\aliha\Desktop\CloudAppDev\AlikhanMurat\Midterm> node -v
v20.18.0
PS C:\Users\aliha\Desktop\CloudAppDev\AlikhanMurat\Midterm> npm -v
10.9.0
```

Now Node.js should be installed on your system. You can check it writing “**node -v**” on your terminal

```
PS C:\Users\aliha\Desktop\CloudAppDev\Midterm> npx create-react-app midterm-project
npm error code ENOENT
npm error path C:\Users\aliha\AppData\Roaming\npm
npm error errno -4058
npm error enoent ENOENT: no such file or directory, lstat 'C:\Users\aliha\AppData\Roaming\npm'
npm error enoent This is related to npm not being able to find a file.
npm error enoent
npm notice
npm notice New minor version of npm available! 10.8.2 -> 10.9.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.9.0
npm notice To update run: npm install -g npm@10.9.0
npm notice
npm error A complete log of this run can be found in: C:\Users\aliha\AppData\Local\npm-cache\_logs\2024-10-17T17_44_27_951Z-debug-0.log
PS C:\Users\aliha\Desktop\CloudAppDev\Midterm> npx create-react-app weather-dashboard
```

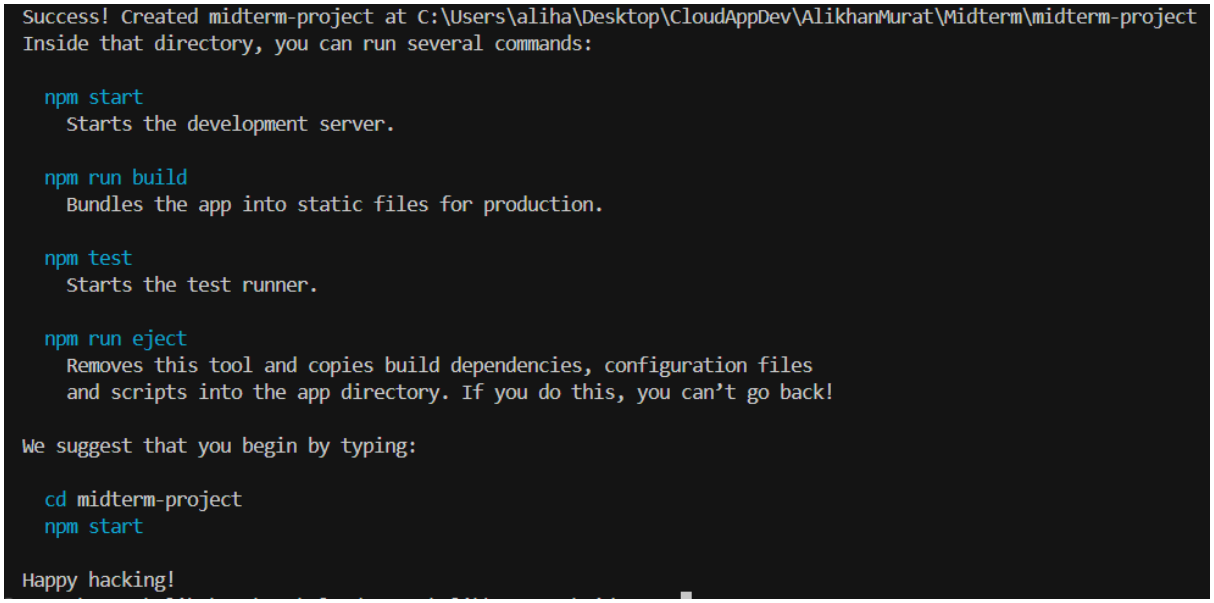
Create a react app with the command "**npx create-react-app project-name**". After entering the code, if you struggle with a message like in the picture above.

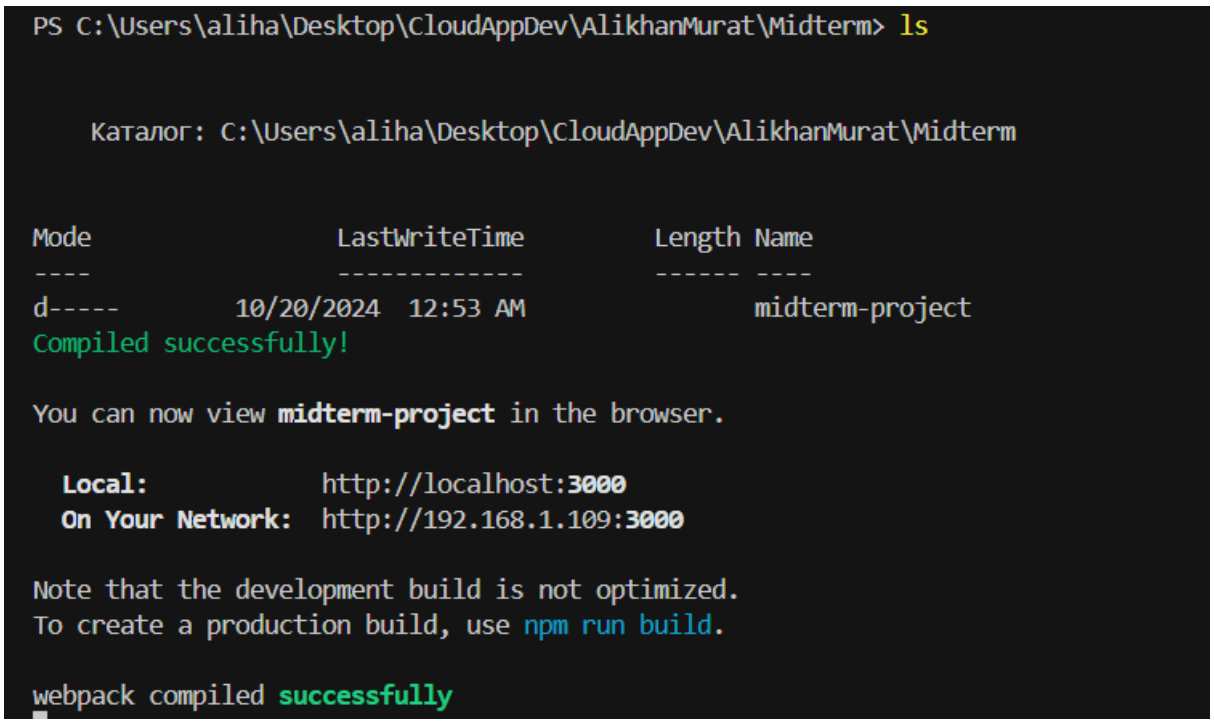
```
PS C:\Users\aliha\Desktop\CloudAppDev\Midterm> npm install -g npm@latest
>>

added 1 package in 7s

25 packages are looking for funding
  run `npm fund` for details
```

Write "**npm install -g npm@latest**" and enter. This installs the latest version of npm globally on your system.

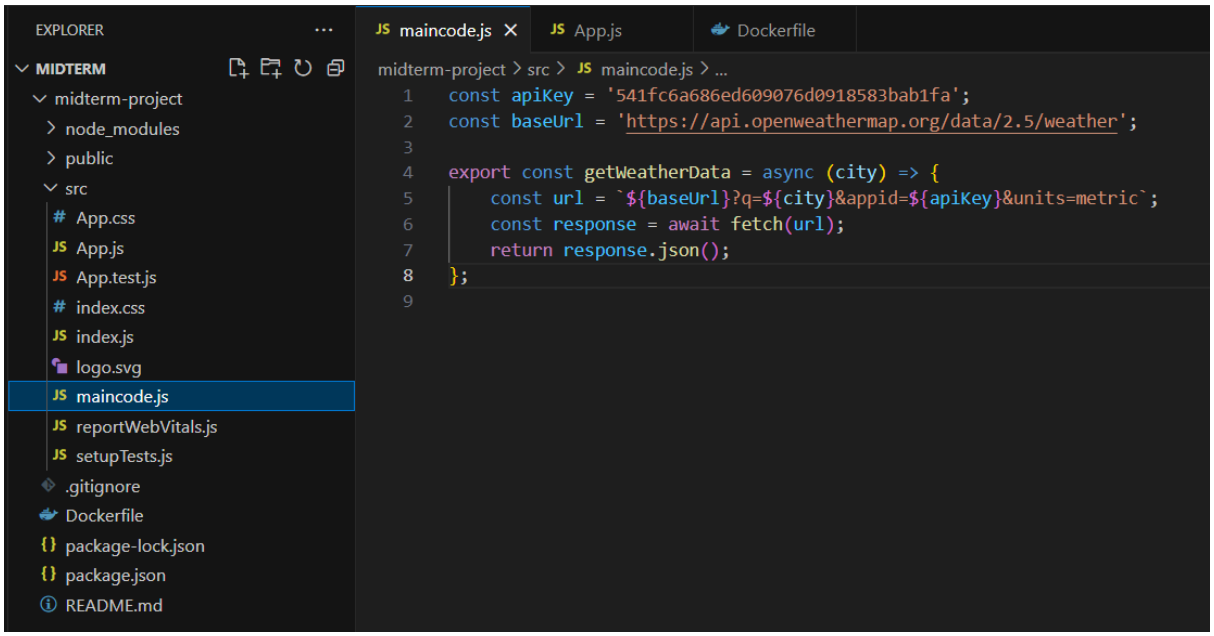
```
PS C:\Users\aliha\Desktop\CloudAppDev\AlikhanMurat\Midterm> npx create-react-app midterm-project

Creating a new React app in C:\Users\aliha\Desktop\CloudAppDev\AlikhanMurat\Midterm\midterm-project.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1478 packages in 2m

262 packages are looking for funding
  run `npm fund` for details

Installing template dependencies using npm...

added 63 packages, and changed 1 package in 12s

262 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...


removed 1 package, and audited 1541 packages in 4s
```

re-enter "**npx create-react-app project-name**"

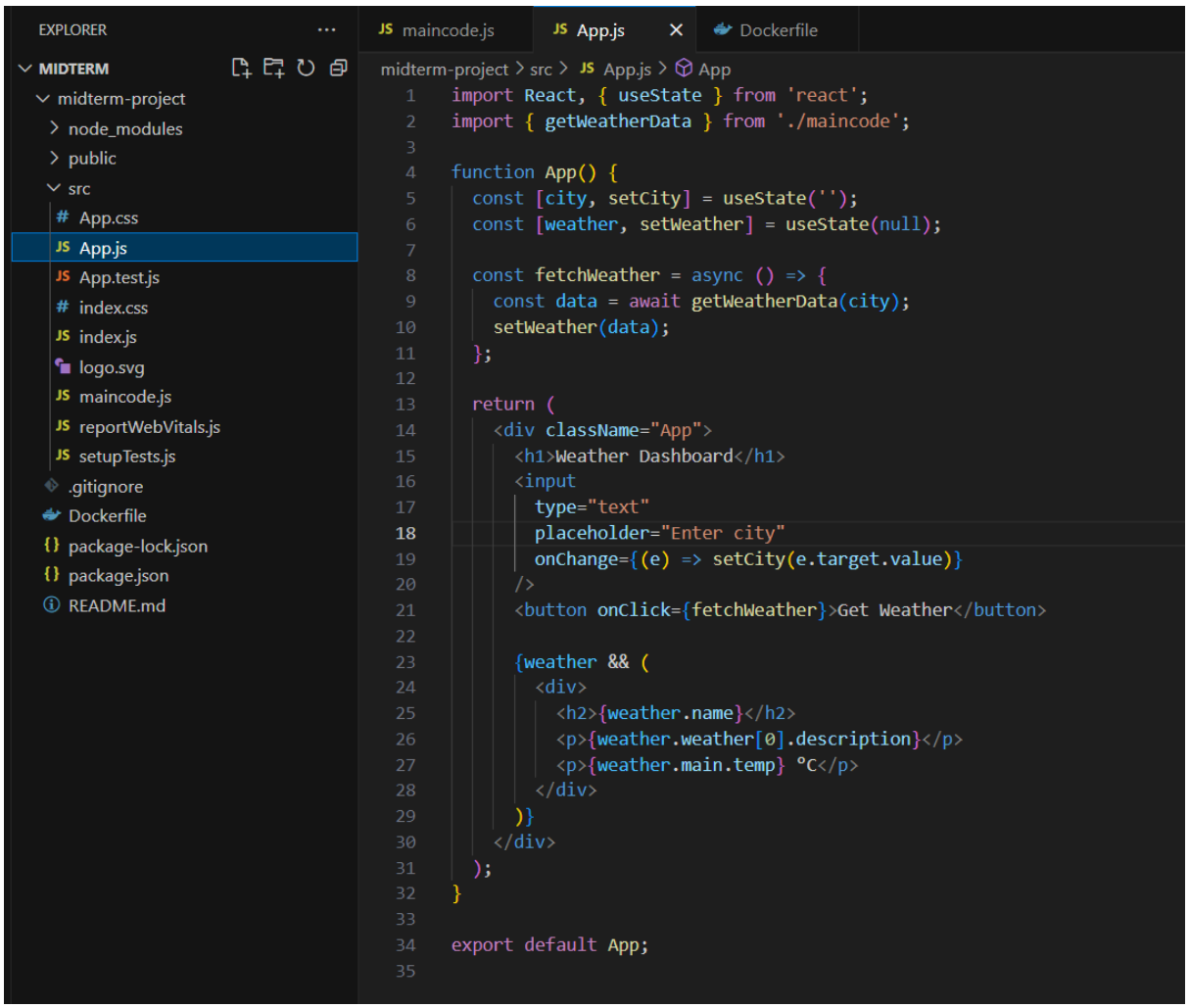




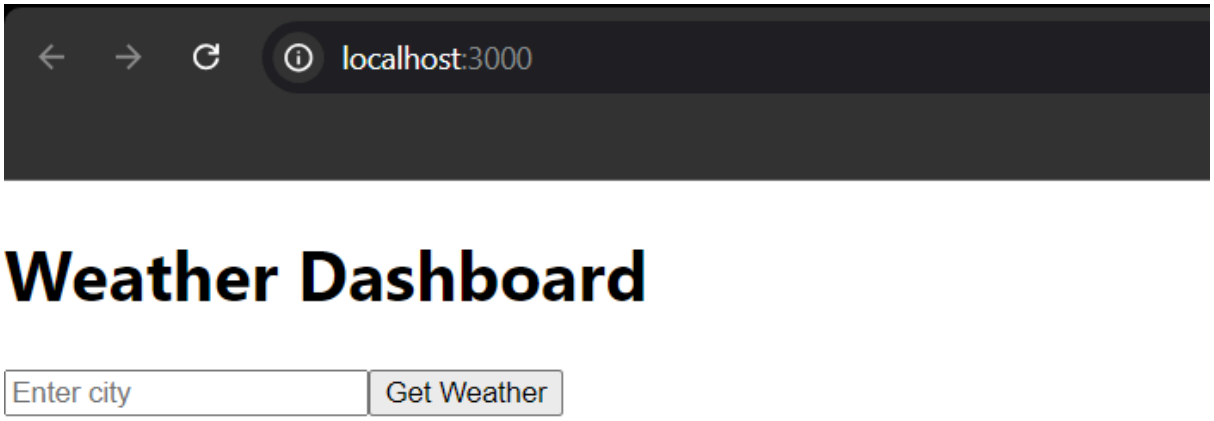
After that, change your directory to your project folder (cd midterm-project) and write "**npm start**". This should initialize your project. You can check you website by the link "**http://localhost:3000**", for now there will be shown only a blank page (react logo).



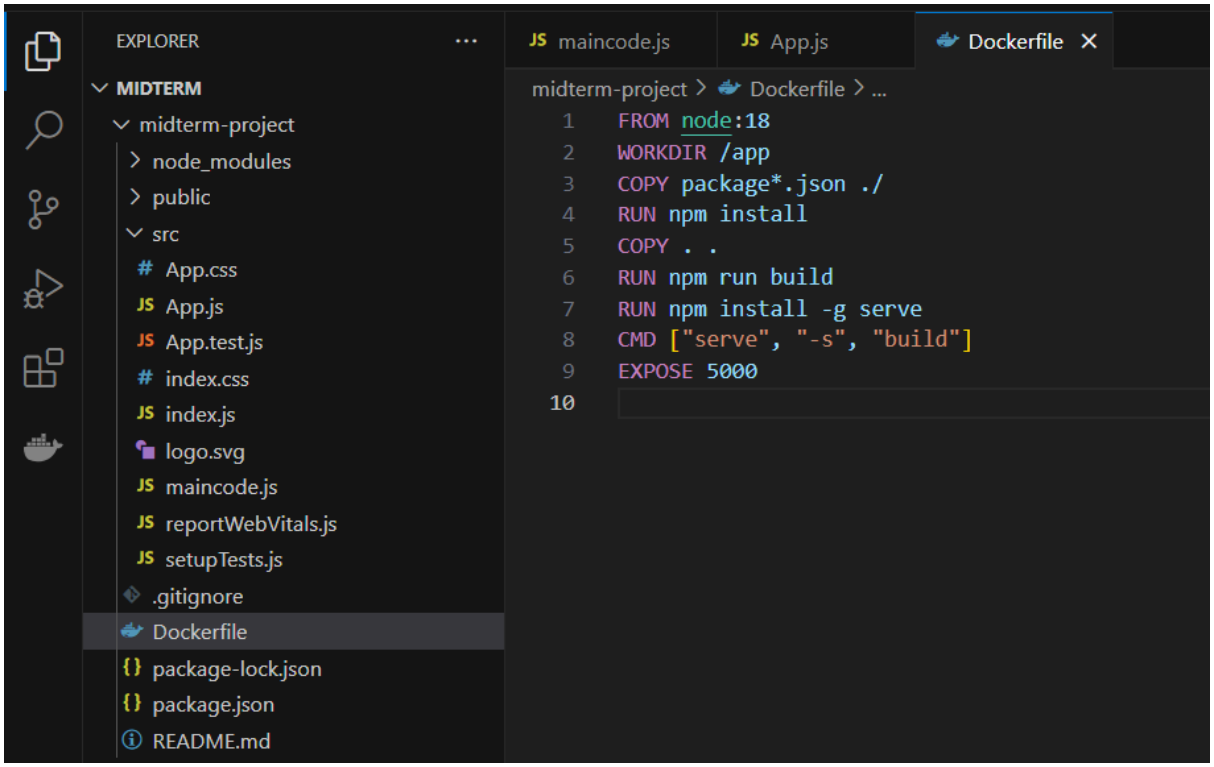
Create a new javascript file where contains **API key** from **OpenWeatherMap** to get information about weather



On **App.js** change the code and import from the file that was previously created. You change the design of the website, in my case I created a simple website (Because I only know beginner type of codes and website designing, since I'm an Automation and Control student)).



Save your code. Now you can check your website. It should look something like the image above.



To make the website public you should deploy it via **Google Cloud Run**. So, first of all, create a **Dockerfile** and write the code that mentioned above and save it.

```
PS C:\Users\aliha\Desktop\CloudAppDev\AlikhanMurat\Midterm\midterm-project> gcloud builds submit --tag gcr.io/cloud-app-dev-midterm/midterm-project
Creating temporary archive of 19 file(s) totalling 761.0 KiB before compression.
Some files were not included in the source upload.

Check the gcloud log [C:\Users\aliha\AppData\Roaming\gcloud\logs\2024.10.20\01.07.27.773871.log] to see which files and the contents of the
default gcloudignore file used (see `$ gcloud topic gcloudignore` to learn
more).

Uploading tarball of [.] to [gs://cloud-app-dev-midterm_cloudbuild/source/1729368447.970845-95616086c3a14a028cdbff32ae8e9fbb.tgz]
Created [https://cloudbuild.googleapis.com/v1/projects/cloud-app-dev-midterm/locations/global/builds/0f1aa785-baaa-44a1-ad25-102940be6051].
Logs are available at [ https://console.cloud.google.com/cloud-build/builds/0f1aa785-baaa-44a1-ad25-102940be6051?project=425509735138 ].
Waiting for build to complete. Polling interval: 1 second(s).
------------------------------------------------------------ REMOTE BUILD OUTPUT ------------------------------------------------------------
starting build "0f1aa785-baaa-44a1-ad25-102940be6051"
```

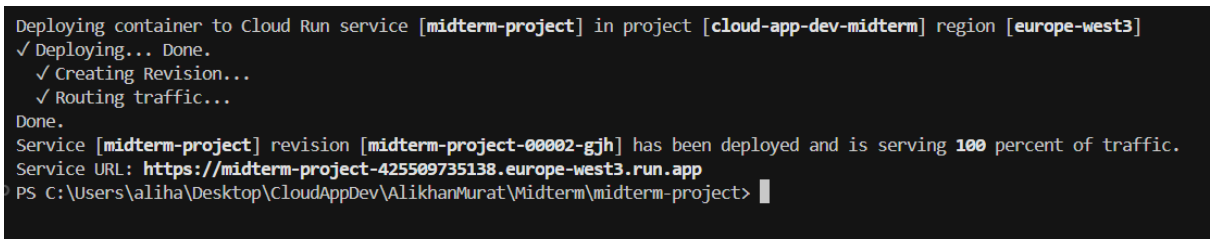
After that on your terminal write "**gcloud builds submit --tag gcr.io/google-project-id/project-name**". This command will build the application using **Google Cloud Build** and store the **Docker image** in **Google Container Registry**. Later Docker image can be deployed to **Cloud Run**.

```
Step 4/9 : RUN npm install
 ---> Running in 0c1028efb688
npm warn deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm warn deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility ta
ble on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm warn deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm warn deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated q@1.5.1: You or someone you depend on is using Q, the JavaScript Promise library that gave JavaScript developers strong feelings
about promises. They can almost certainly migrate to the native JavaScript promise now. Thank you literally everyone for joining me in this bet again
st the odds. Be excellent to each other.
npm warn deprecated
npm warn deprecated (For a CapTP with native promises, see @endo/eventual-send and @endo/captp)
npm warn deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm warn deprecated workbox-google-analytics@6.6.0: It is not compatible with newer versions of GA starting with v4, as long as you are using GAv3 it
 should be ok, but the package is not longer being maintained
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested
way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated domexception@2.0.1: Use your platform's native DOMException instead
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated @humanwhocodes/object-schema@2.0.3: Use @eslint/object-schema instead
npm warn deprecated @humanwhocodes/config-array@0.13.0: Use @eslint/config-array instead
npm warn deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
npm warn deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is
 no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm warn deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is
 no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm warn deprecated @babel/plugin-proposal-private-methods@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is n
o longer maintained. Please use @babel/plugin-transform-private-methods instead.
npm warn deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is
no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm warn deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this
 plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm warn deprecated eslint@8.57.1: This version is no longer supported. Please see https://eslint.org/version-support for other options.
```

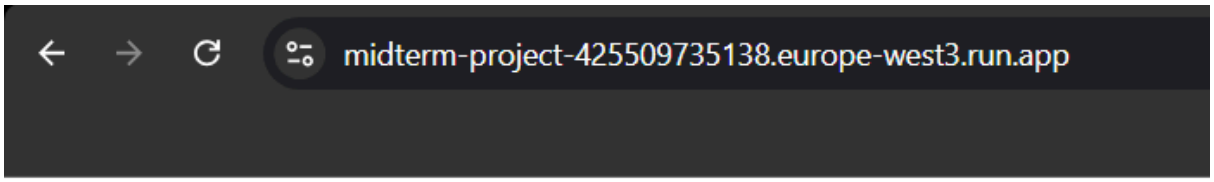
This can be seen as an error, but it's okay. It's the part of the process)

```
 [cloud-app-dev-midterm]. Permission must be granted to the Google Cloud Run Service Agent service-425509735138@serverless-robot-prod.iam.gserviceacc
ount.com from this project. See https://cloud.google.com/run/docs/deploying#gcloud run deploy midterm-project --image gcr.io/cloud-app-dev-midterm/mi
dterm-project --platform managedppDev\AlikhanMurat\Midterm\midterm-project>
Please specify a region:
 [1] africa-south1
 [2] asia-east1
 [3] asia-east2
 [4] asia-northeast1
 [5] asia-northeast2
 [6] asia-northeast3
 [7] asia-south1
 [8] asia-south2
 [9] asia-southeast1
 [10] asia-southeast2
 [11] australia-southeast1
 [12] australia-southeast2
 [13] europe-central2
 [14] europe-north1
 [15] europe-southwest1
 [16] europe-west1
 [17] europe-west10
 [18] europe-west12
 [19] europe-west2
 [20] europe-west3
 [21] europe-west4
 [22] europe-west6
 [23] europe-west8
 [24] europe-west9
 [25] me-central1
 [26] me-central2
 [27] me-west1
 [28] northamerica-northeast1
 [29] northamerica-northeast2
 [30] southamerica-east1
 [31] southamerica-west1
 [32] us-central1
```

After everything, write the command "**gcloud run deploy storing-name --image gcr.io/google-project-id/project-name --platform managed**" and choose the region, in my case "**europe-west3**" , the most convenient and optimal one. This command deploys containerized application to **Google Cloud Run**



Congratulations, your web application is now ready!



Weather Dashboard

Almaty Get Weather

Almaty

smoke

5.95 °C

https://midterm-project-425509735138.europe-west3.run.app/

Conclusion

In conclusion, The Weather Dashboard project successfully achieved its goal by using GCP's infrastructure,tools and services. The application provides a user-friendly experience to retrieve current weather conditions, achieving high availability, scalability, and performance. In future, improvements may include adding user accounts to save favorite cities, advanced weather analytics and improving the user interface. (I'm an Automation and Control student, so I tried my best. If there are mistakes , don't judge me.)))

References

https://www.geeksforgeeks.org/weather-application-using-reactjs/
https://smartshock.hashnode.dev/create-a-weather-app-with-react-a-step-by-step-guide
https://www.youtube.com/watch?v=zs1Nq2s_uy4
https://helpdeskgeek.com/windows-10/add-windows-path-environment-variable/
https://github.com/NereidaRondon/weather-dashboard
https://docs.google.com/presentation/d/13gd8z1plqIm5rGGPRgeAm4Ju6_W0mH6C/edit?classId=dd8088fe-69ff-4a1d-b89e-d776c0f63ba2#slide=id.p2