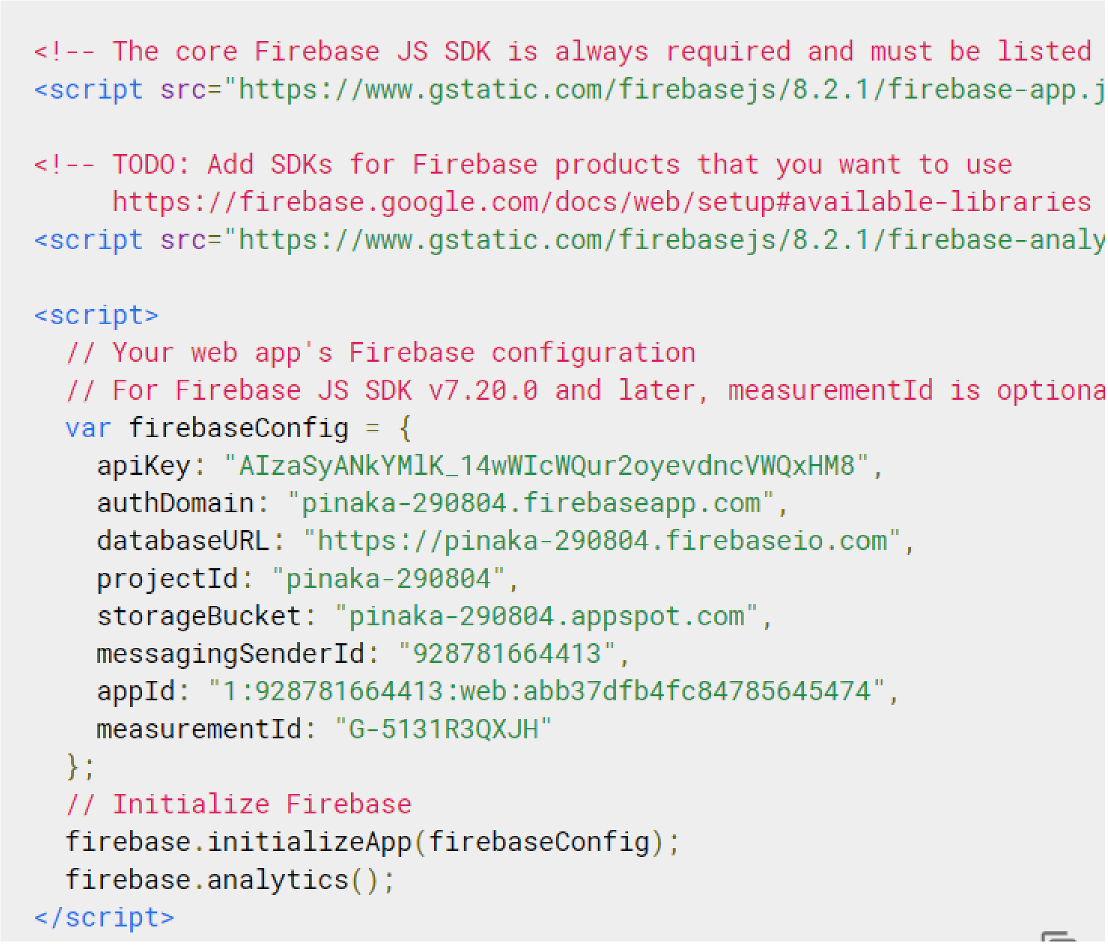
Pinaka Web App
Code manual

Setup Variables in Pinaka Web App:

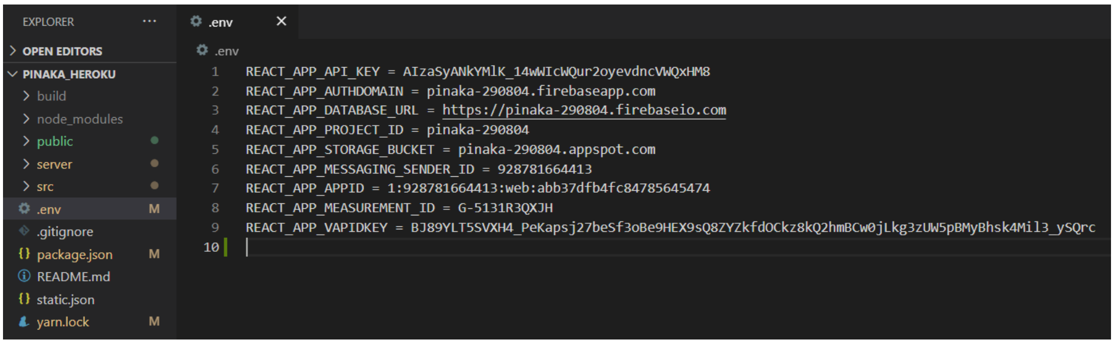
1.

Change the Firebase config in the .env file :

1. https://firebase.google.com/
2. Go to console
3. Choose your firebase project
4. Click on project setting
5. Copy "firebaseConfig" to .env file in our project root



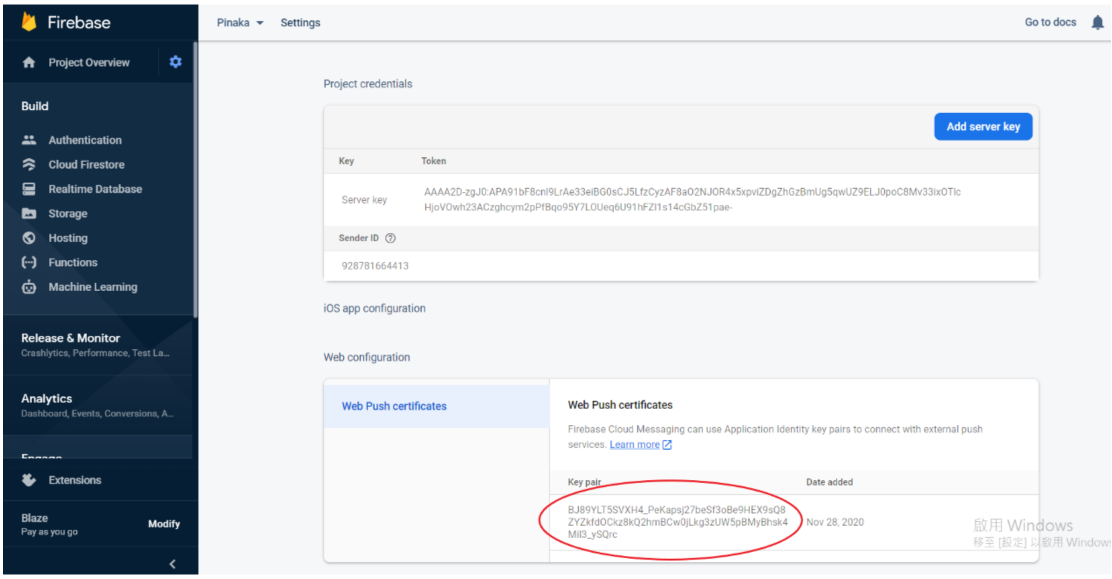
```
<!-- The core Firebase JS SDK is always required and must be listed
<script src="https://www.gstatic.com/firebasejs/8.2.1/firebase-app.j

<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries
<script src="https://www.gstatic.com/firebasejs/8.2.1/firebase-analy

<script>
  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optiona
  var firebaseConfig = {
    apiKey: "AIzaSyANkYMlK_14wWIcWQur2oyevdncVWQxHM8",
    authDomain: "pinaka-290804.firebaseapp.com",
    databaseURL: "https://pinaka-290804.firebaseio.com",
    projectId: "pinaka-290804",
    storageBucket: "pinaka-290804.appspot.com",
    messagingSenderId: "928781664413",
    appId: "1:928781664413:web:abb37dfb4fc84785645474",
    measurementId: "G-5131R3QXJH"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```



```
REACT_APP_API_KEY = AIzaSyANkYMlK_14wWIcWQur2oyevdncVWQxHM8
REACT_APP_AUTHDOMAIN = pinaka-290804.firebaseapp.com
REACT_APP_DATABASE_URL = https://pinaka-290804.firebaseio.com
REACT_APP_PROJECT_ID = pinaka-290804
REACT_APP_STORAGE_BUCKET = pinaka-290804.appspot.com
REACT_APP_MESSAGING_SENDER_ID = 928781664413
REACT_APP_APPID = 1:928781664413:web:abb37dfb4fc84785645474
REACT_APP_MEASUREMENT_ID = G-5131R3QXJH
REACT_APP_VAPIDKEY = BJ89YLT5SVXH4_PeKapsj27beSf3oBe9HEX9sQ8ZYZkfdOCkz8kQ2hmBCw0jLkg3zUW5pBMyBhsk4Mil3_ySQrc
```

VAPIDKEY:

1. https://firebase.google.com/
2. Go to console
3. Choose your firebase project
4. Click on project setting
5. Click on cloud messaging
6. Copy VAPIDKEY below (Create a new VAPIDKEY if you don't have)



Project credentials

Key	Token
Server key	AAAA2D-zgJ0:APA91bF8cnl9LrAe33eiBG0sCJ5LfzCyzAF8aO2NJOR4x5xpviZDgZhGzBmUg5qwUZ9ELJ0poC8Mv33ixOTIc HjoVOwh23ACzghcym2pPfBqo95Y7LOUeq6U91hFZI1s14cGbZ51pae-
Sender ID	
928781664413	

iOS app configuration

Web configuration

Web Push certificates

Web Push certificates

Firebase Cloud Messaging can use Application Identity key pairs to connect with external push services. Learn more

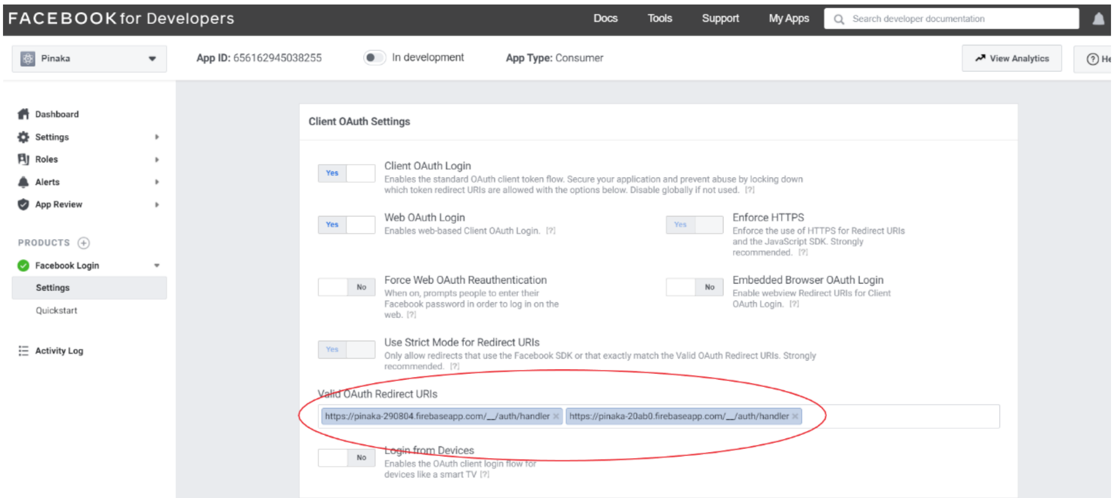
Key pair	Date added
BJ89YLT5SVXH4_PeKapsj27beSf3oBe9HEX9sQ8ZYZkfdOCkz8kQ2hmBCw0jLkg3zUW5pBMyBhsk4Mil3_ySQrc	Nov 28, 2020

7. Change the VAPIDKEY to yours in .env just like above

2.

Setup Facebook login functionality

1. https://developers.facebook.com/
2. Go to “My Apps”
3. Click on your project
4. Click on “Facebook Login -> Settings” in the sidebar
5. Turn on “Enfore HTTPS”
6. And add URIs below:



App ID: 656162945038255 In development App Type: Consumer

Client OAuth Settings

Yes Client OAuth Login
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

Yes Web OAuth Login
Enables web-based Client OAuth Login. [?]

Yes Enforce HTTPS
Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. [?]

No Force Web OAuth Reauthentication
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

No Embedded Browser OAuth Login
Enable webview Redirect URIs for Client OAuth Login. [?]

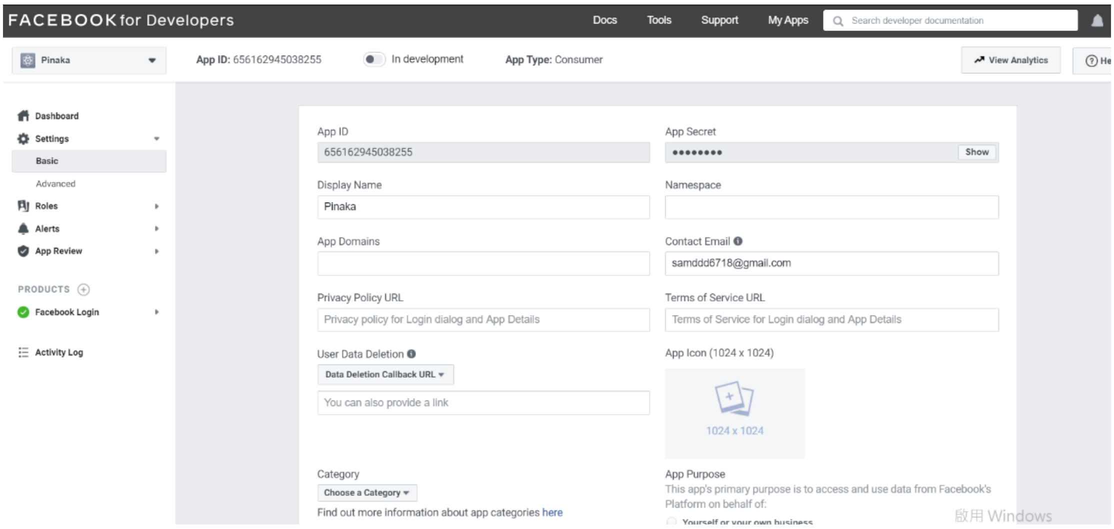
Yes Use Strict Mode for Redirect URIs
Only allow redirects that use the Facebook SDK or that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]

Valid OAuth Redirect URIs
https://pinaka-290804.firebaseapp.com/__/auth/handler × https://pinaka-20ab0.firebaseapp.com/__/auth/handler ×

No Login from Devices
Enables the OAuth client login flow for devices like a smart TV [?]

Where to get the URI:

1. Go to firebase console as above
2. Click on Authentication on the sidebar
3. Click on Sign-in method
4. Click on Facebook below
5. Add your App ID and App secret in your Facebook Developers to your firebase project as the images in point 6



App ID: 656162945038255 In development App Type: Consumer

App ID: 656162945038255

App Secret: ••••••••

Display Name: Pinaka

Namespace:

App Domains:

Contact Email: samddd6718@gmail.com

Privacy Policy URL: Privacy policy for Login dialog and App Details

Terms of Service URL: Terms of Service for Login dialog and App Details

User Data Deletion

You can also provide a link

App Icon (1024 x 1024)

Category

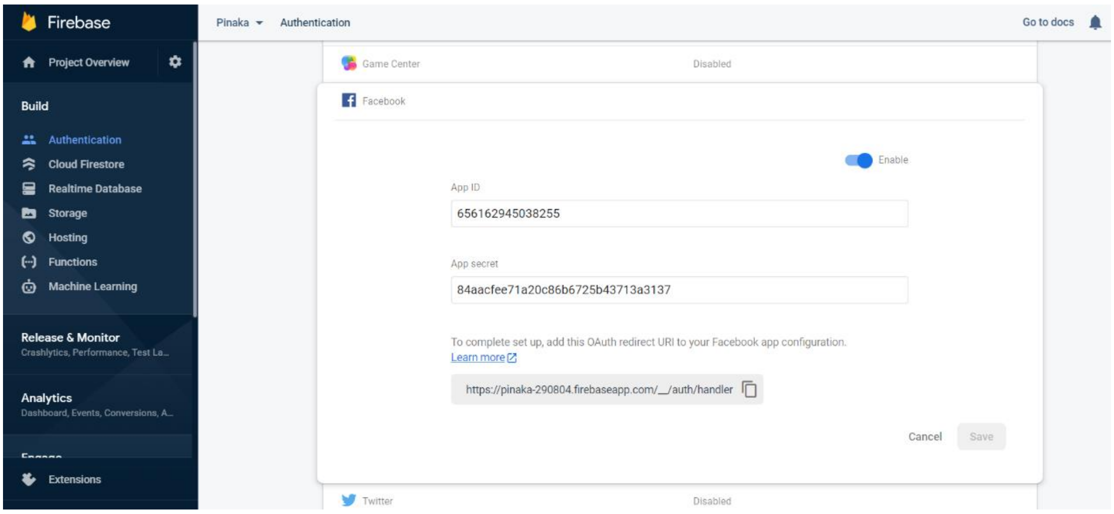
Find out more information about app categories here

App Purpose

This app's primary purpose is to access and use data from Facebook's Platform on behalf of:

Yourself or your own business

6. Copy the URL below:



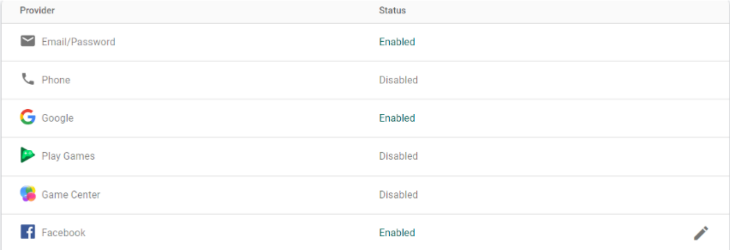
App ID: 656162945038255

App secret: 84aacfee71a20c86b6725b43713a3137

To complete set up, add this OAuth redirect URI to your Facebook app configuration.

https://pinaka-290804.firebaseapp.com/__/auth/handler

Please also enable these:

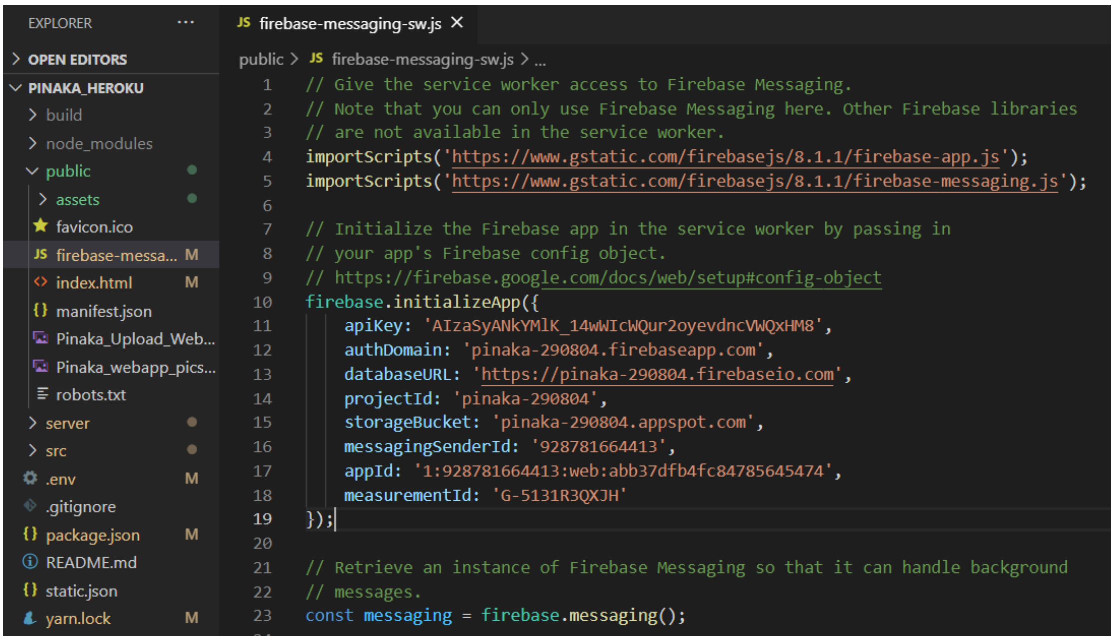


Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Enabled
Play Games	Disabled
Game Center	Disabled
Facebook	Enabled

2.

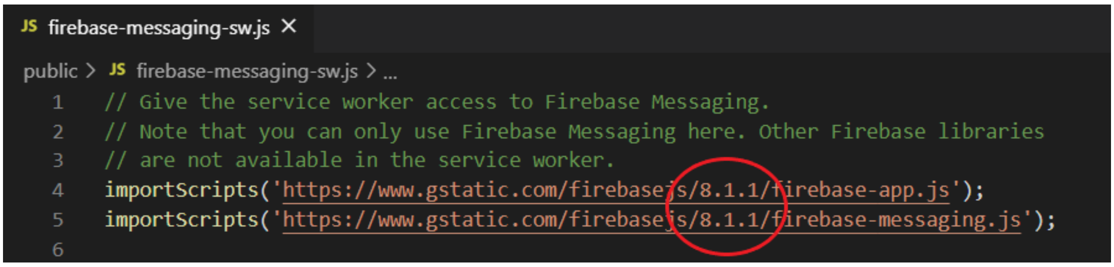
Copy the same “firebaseConfig” to firebase-messaging-sw:

1. Go to public directory
2. Open firebase-messaging-sw file
3. Change the content of firebase.initializeApp to your “firebaseConfig” above



```
// Give the service worker access to Firebase Messaging.
// Note that you can only use Firebase Messaging here. Other Firebase libraries
// are not available in the service worker.
importScripts('https://www.gstatic.com/firebasejs/8.1.1/firebase-app.js');
importScripts('https://www.gstatic.com/firebasejs/8.1.1/firebase-messaging.js');

// Initialize the Firebase app in the service worker by passing in
// your app's Firebase config object.
// https://firebase.google.com/docs/web/setup#config-object
firebase.initializeApp({
    apiKey: 'AIzaSyANkYMlK_14wWIcWQur2oyevdncVWQxHM8',
    authDomain: 'pinaka-290804.firebaseapp.com',
    databaseURL: 'https://pinaka-290804.firebaseio.com',
    projectId: 'pinaka-290804',
    storageBucket: 'pinaka-290804.appspot.com',
    messagingSenderId: '928781664413',
    appId: '1:928781664413:web:abb37dfb4fc84785645474',
    measurementId: 'G-5131R3QXJH'
});

// Retrieve an instance of Firebase Messaging so that it can handle background
// messages.
const messaging = firebase.messaging();
```

Make sure the version should be up-to-date for working FCM:

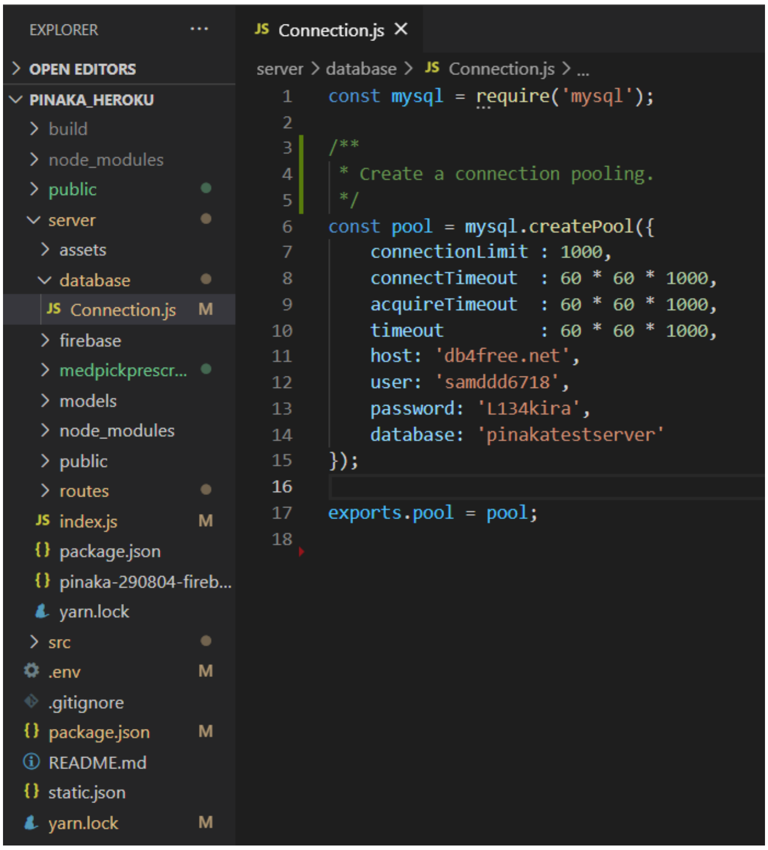


```
// Give the service worker access to Firebase Messaging.
// Note that you can only use Firebase Messaging here. Other Firebase libraries
// are not available in the service worker.
importScripts('https://www.gstatic.com/firebasejs/8.1.1/firebase-app.js');
importScripts('https://www.gstatic.com/firebasejs/8.1.1/firebase-messaging.js');

```

3.

Database connection:

1. Go to “server/database/Connection.js”
2. Please change host, user, password, and database as your database information.



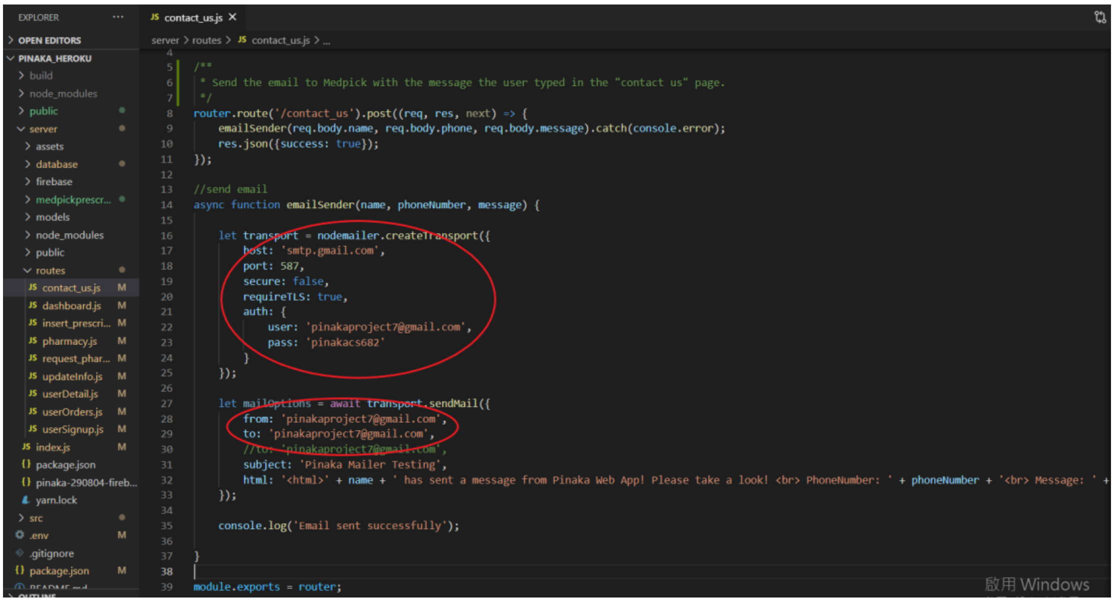
```
const mysql = require('mysql');

/**
 * Create a connection pooling.
 */
const pool = mysql.createPool({
    connectionLimit : 1000,
    connectTimeout  : 60 * 60 * 1000,
    acquireTimeout  : 60 * 60 * 1000,
    timeout         : 60 * 60 * 1000,
    host: 'db4free.net',
    user: 'samddd6718',
    password: 'L134kira',
    database: 'pinakatestserver'
});

exports.pool = pool;

```

4.

Change your email information in these files:

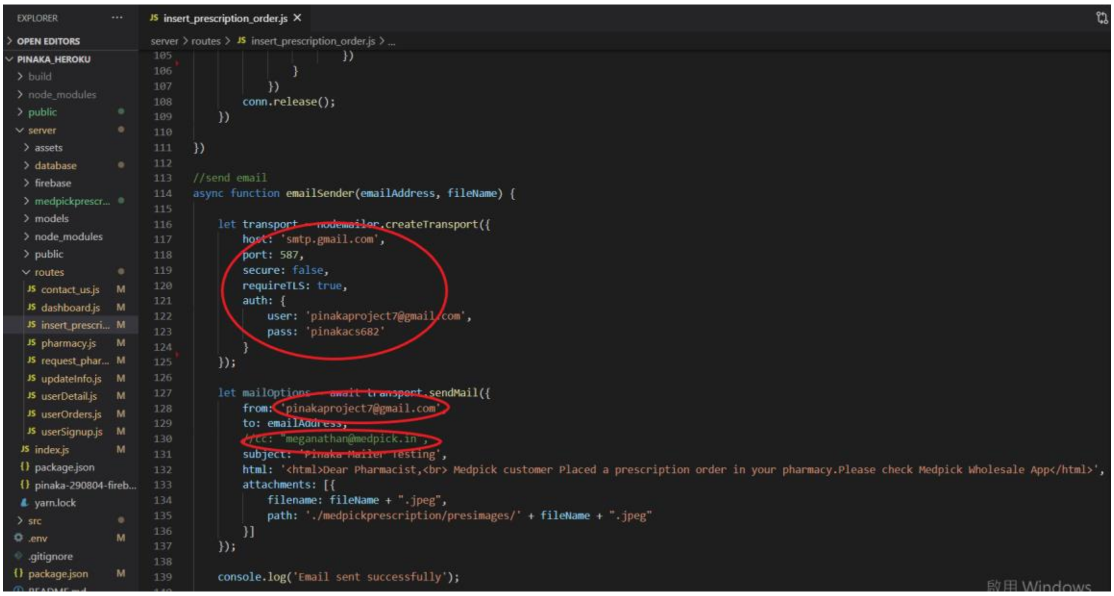
/server/routes/contact_us.js

(“to”: below please set to the email address you want to receive from the customer’s message)



```
/**
 * Send the email to Medpick with the message the user typed in the "contact us" page.
 */
router.route('/contact_us').post((req, res, next) => {
    emailSender(req.body.name, req.body.phone, req.body.message).catch(console.error);
    res.json({success: true});
});

//send email
async function emailSender(name, phoneNumber, message) {

    let transport = nodemailer.createTransport({
        host: 'smtp.gmail.com',
        port: 587,
        secure: false,
        requireTLS: true,
        auth: {
            user: 'pinakaproject7@gmail.com',
            pass: 'pinakacs682'
        }
    });

    let mailOptions = await transport.sendMail({
        from: 'pinakaproject7@gmail.com',
        to: 'pinakaproject7@gmail.com',
        //to: 'pinakaproject7@gmail.com',
        subject: 'Pinaka Mailer Testing',
        html: '<html>' + name + ' has sent a message from Pinaka Web App! Please take a look! <br> PhoneNumber: ' + phoneNumber + '<br> Message: ' +
    });

    console.log('Email sent successfully');

}

module.exports = router;
```

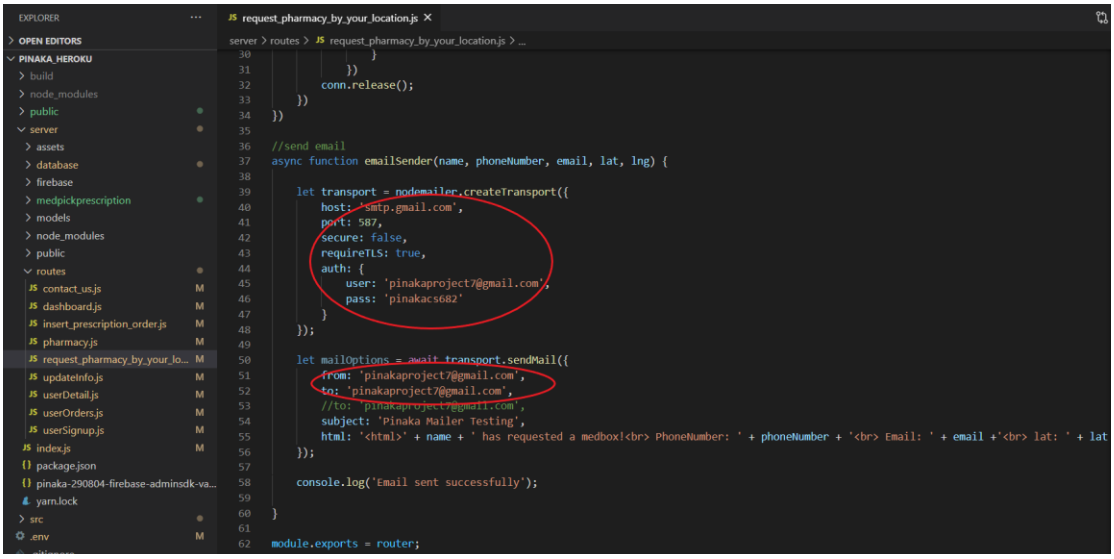
/server/routes/insert_prescription_order.js



```
                })
            }
        })
        conn.release();
    })

})

//send email
async function emailSender(emailAddress, fileName) {

    let transport = nodemailer.createTransport({
        host: 'smtp.gmail.com',
        port: 587,
        secure: false,
        requireTLS: true,
        auth: {
            user: 'pinakaproject7@gmail.com',
            pass: 'pinakacs682'
        }
    });

    let mailOptions = await transport.sendMail({
        from: 'pinakaproject7@gmail.com',
        to: emailAddress,
        //cc: "meganathan@medpick.in",
        subject: 'Pinaka Mailer Testing',
        html: '<html>Dear Pharmacist,<br> Medpick customer Placed a prescription order in your pharmacy.Please check Medpick Wholesale App</html>',
        attachments: [{
            filename: fileName + ".jpeg",
            path: './medpickprescription/presimages/' + fileName + ".jpeg"
        }]
    });

    console.log('Email sent successfully');
```

/server/routes/request_pharmacy_by_your_location.js

(“to”: below please set to the email address you want to receive from the customer’s location in Medbox page)



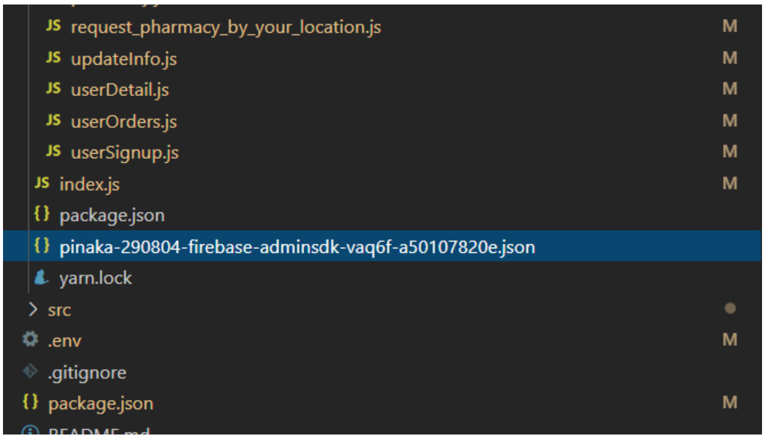
```
                }
            })
        conn.release();
    })
})

//send email
async function emailSender(name, phoneNumber, email, lat, lng) {

    let transport = nodemailer.createTransport({
        host: 'smtp.gmail.com',
        port: 587,
        secure: false,
        requireTLS: true,
        auth: {
            user: 'pinakaproject7@gmail.com',
            pass: 'pinakacs682'
        }
    });

    let mailOptions = await transport.sendMail({
        from: 'pinakaproject7@gmail.com',
        to: 'pinakaproject7@gmail.com',
        //to: 'pinakaproject7@gmail.com',
        subject: 'Pinaka Mailer Testing',
        html: '<html>' + name + ' has requested a medbox!<br> PhoneNumber: ' + phoneNumber + '<br> Email: ' + email +'<br> lat: ' + lat
    });

    console.log('Email sent successfully');

}

module.exports = router;
```

5.

Re-install Firebase Admin SDK to the server with the process in the URL below:

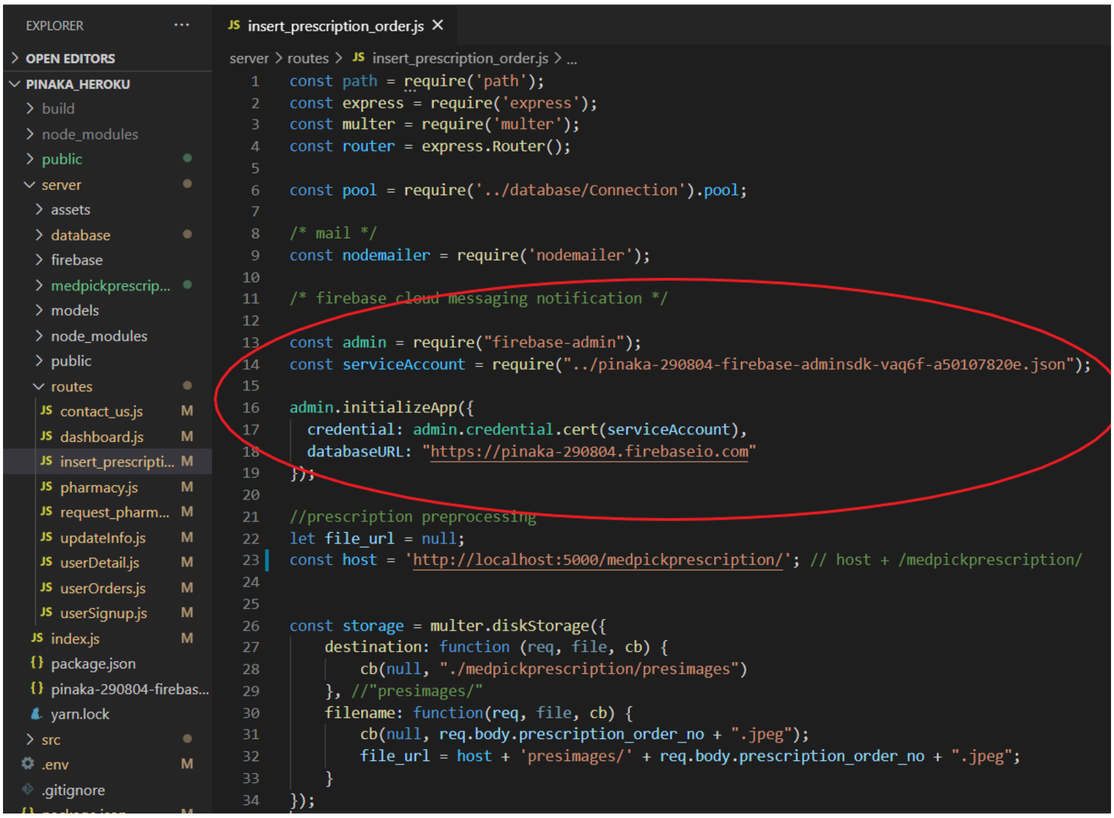
https://firebase.google.com/docs/admin/setup

And it will generate a json file in your server root.



Change the admin information to yours in

/server/routes/insert_prescription_order.js like below:



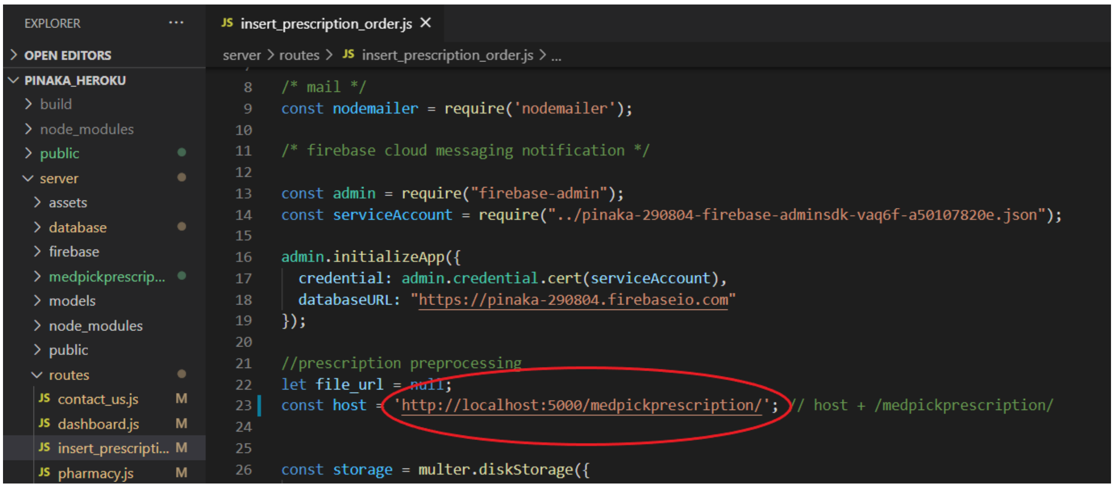
```
const path = require('path');
const express = require('express');
const multer = require('multer');
const router = express.Router();

const pool = require('../database/Connection').pool;

/* mail */
const nodemailer = require('nodemailer');

/* firebase cloud messaging notification */

const admin = require("firebase-admin");
const serviceAccount = require("../pinaka-290804-firebase-adminsdk-vaq6f-a50107820e.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://pinaka-290804.firebaseio.com"
});

//prescription preprocessing
let file_url = null;
const host = 'http://localhost:5000/medpickprescription/'; // host + /medpickprescription/


const storage = multer.diskStorage({
    destination: function (req, file, cb) {
        cb(null, "./medpickprescription/presimages")
    }, //"presimages/"
    filename: function(req, file, cb) {
        cb(null, req.body.prescription_order_no + ".jpeg");
        file_url = host + 'presimages/' + req.body.prescription_order_no + ".jpeg";
    }
});
```

6.

Change URL to our web app's URL in

/server/routes/insert_prescription_order.js

(The form should be like: host_url + /medpickprescription/)



```
/* mail */
const nodemailer = require('nodemailer');

/* firebase cloud messaging notification */

const admin = require("firebase-admin");
const serviceAccount = require("../pinaka-290804-firebase-adminsdk-vaq6f-a50107820e.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://pinaka-290804.firebaseio.com"
});

//prescription preprocessing
let file_url = null;
const host = 'http://localhost:5000/medpickprescription/'; // host + /medpickprescription/


const storage = multer.diskStorage({
```

Start the Web App:

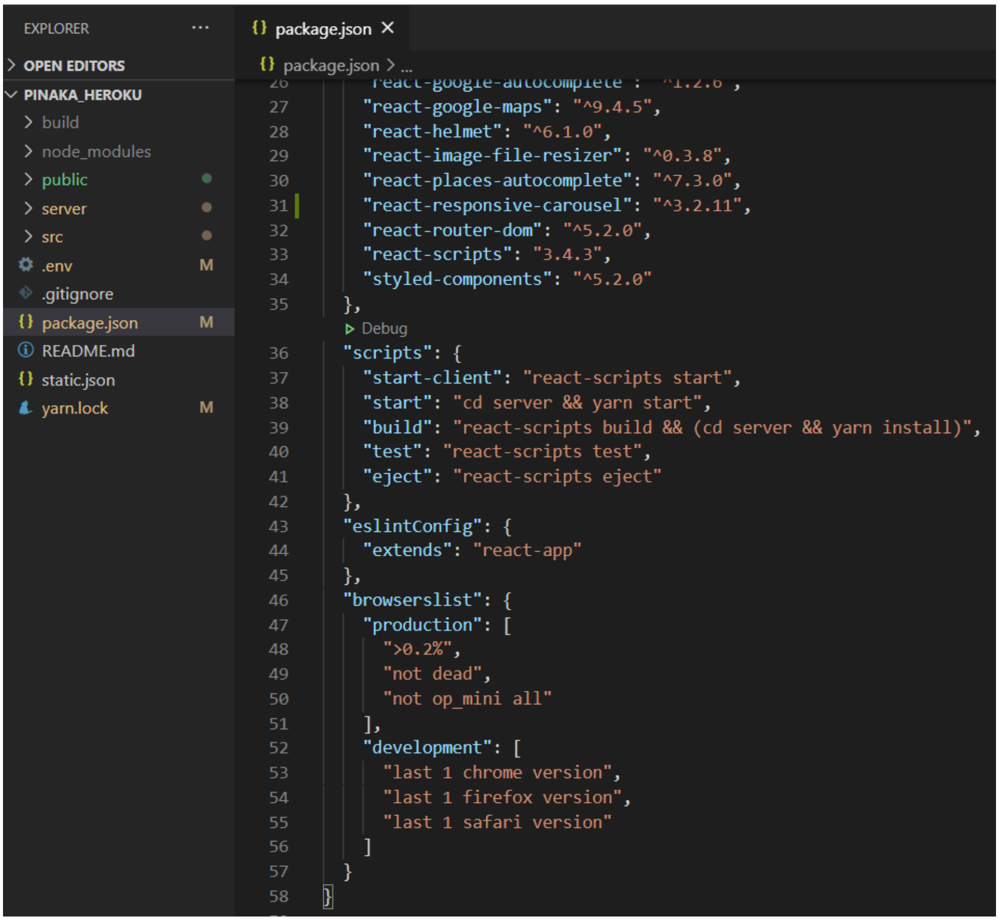
Start: npm start or yarn start

(before start the web app, please execute build : npm build or yarn build)

Local: enter “localhost:5000” in your browser after starting the web app.

Host:

The “start” should be enough for you to use when deploying the web app to your host.



```
    "react-google-autocomplete": "^1.2.6",
    "react-google-maps": "^9.4.5",
    "react-helmet": "^6.1.0",
    "react-image-file-resizer": "^0.3.8",
    "react-places-autocomplete": "^7.3.0",
    "react-responsive-carousel": "^3.2.11",
    "react-router-dom": "^5.2.0",
    "react-scripts": "3.4.3",
    "styled-components": "^5.2.0"
  },
  "scripts": {
    "start-client": "react-scripts start",
    "start": "cd server && yarn start",
    "build": "react-scripts build && (cd server && yarn install)",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```