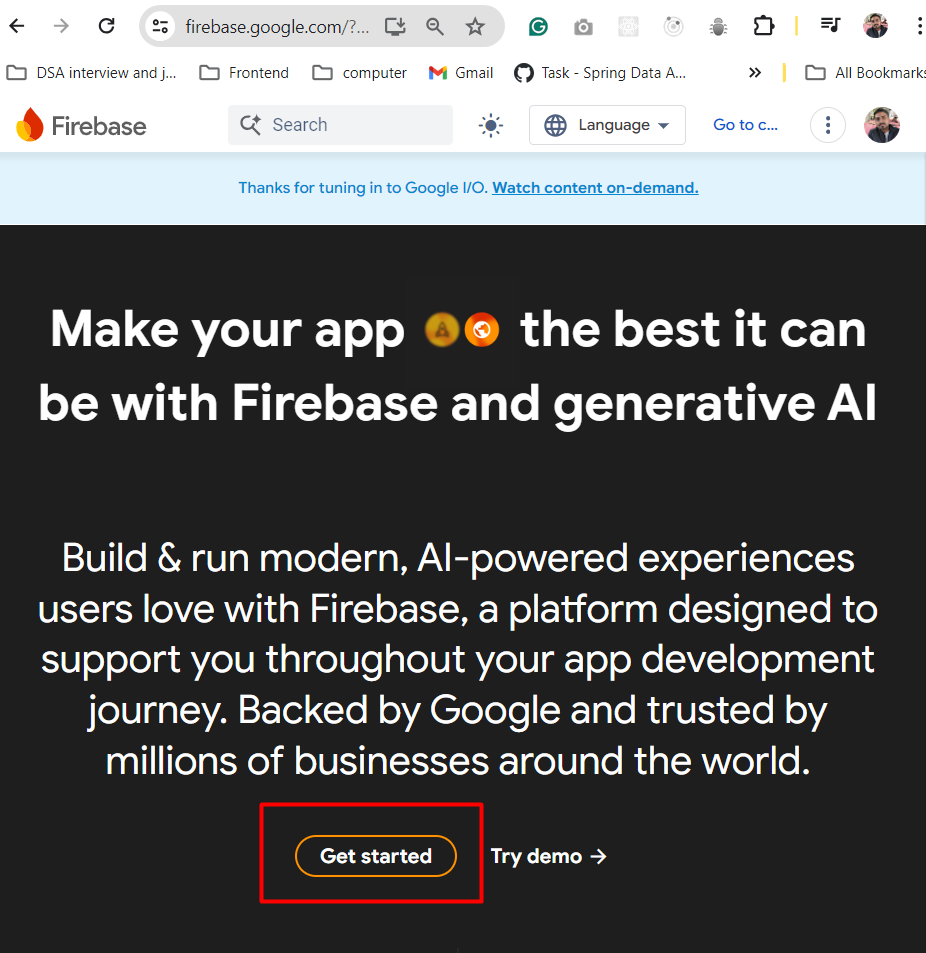
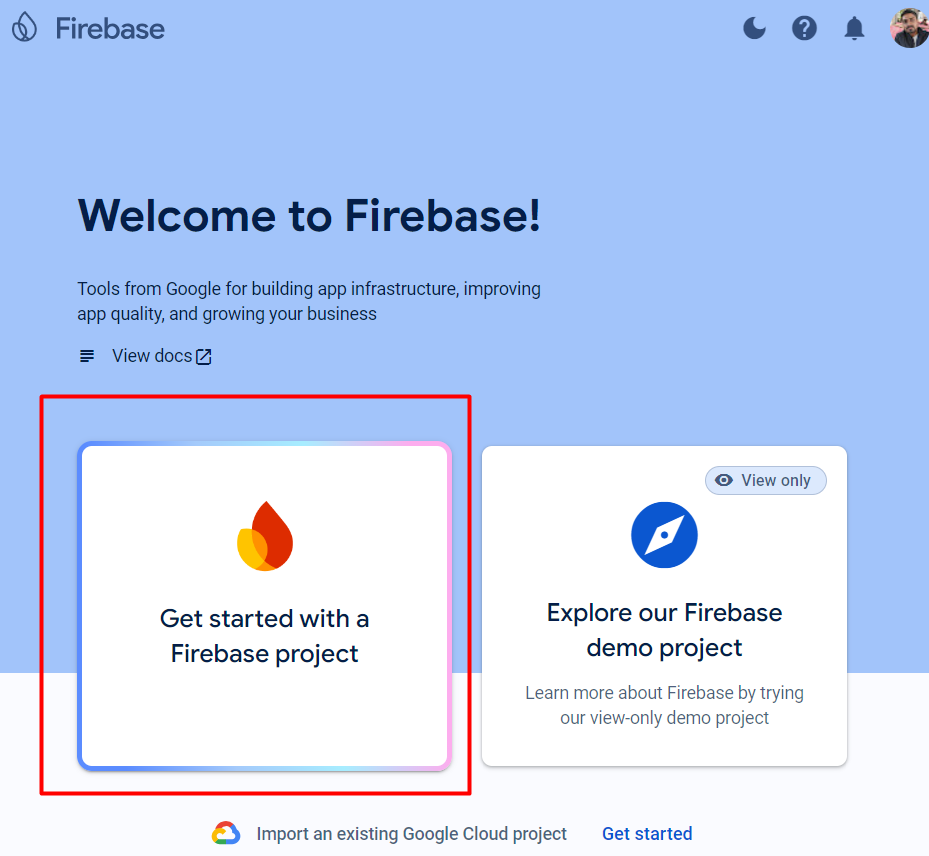
**Push Notification with React js**

**Creating Firebase Google messaging account:**

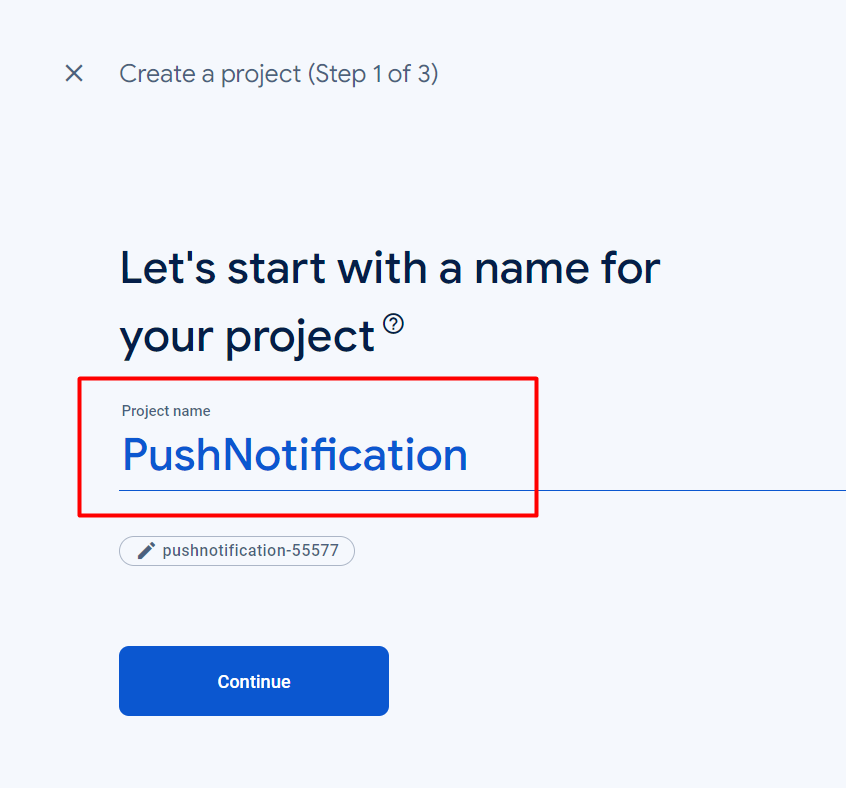
* Step 1: Open firebase profile and click on get started:



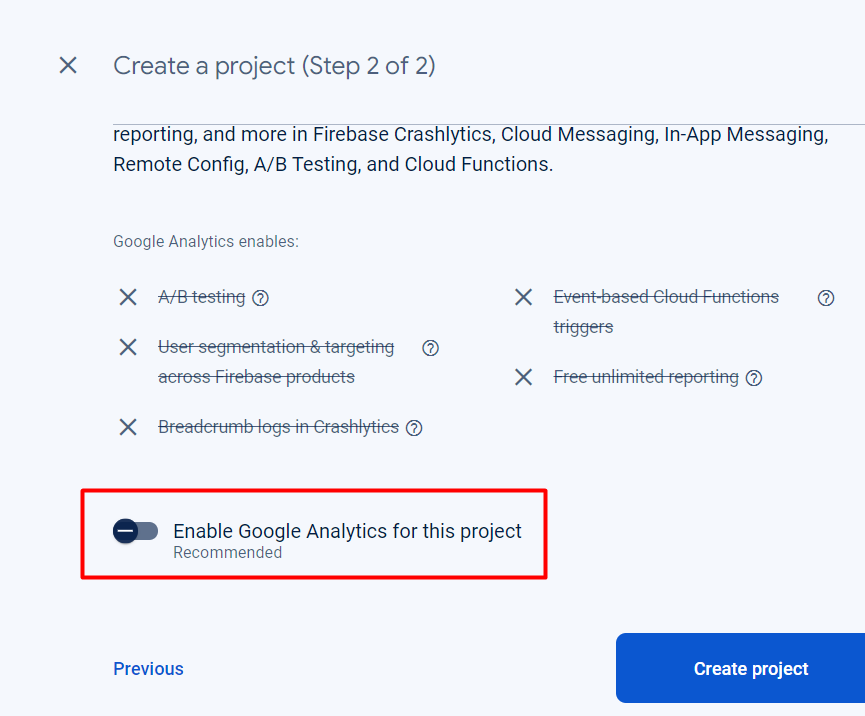
* Step 2: Click on Get started with a firebase project and create new project



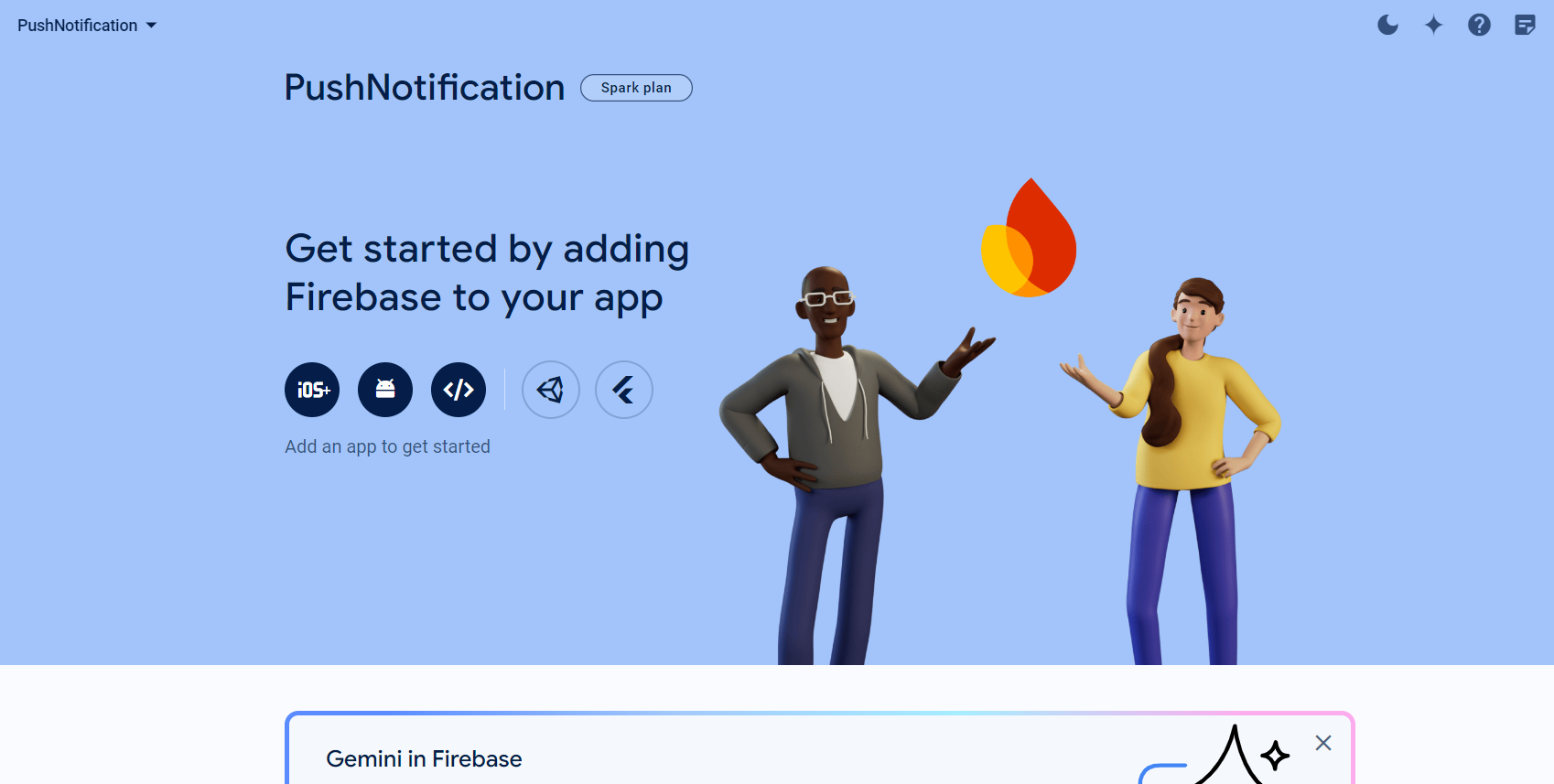
* Step 3: Enter project name



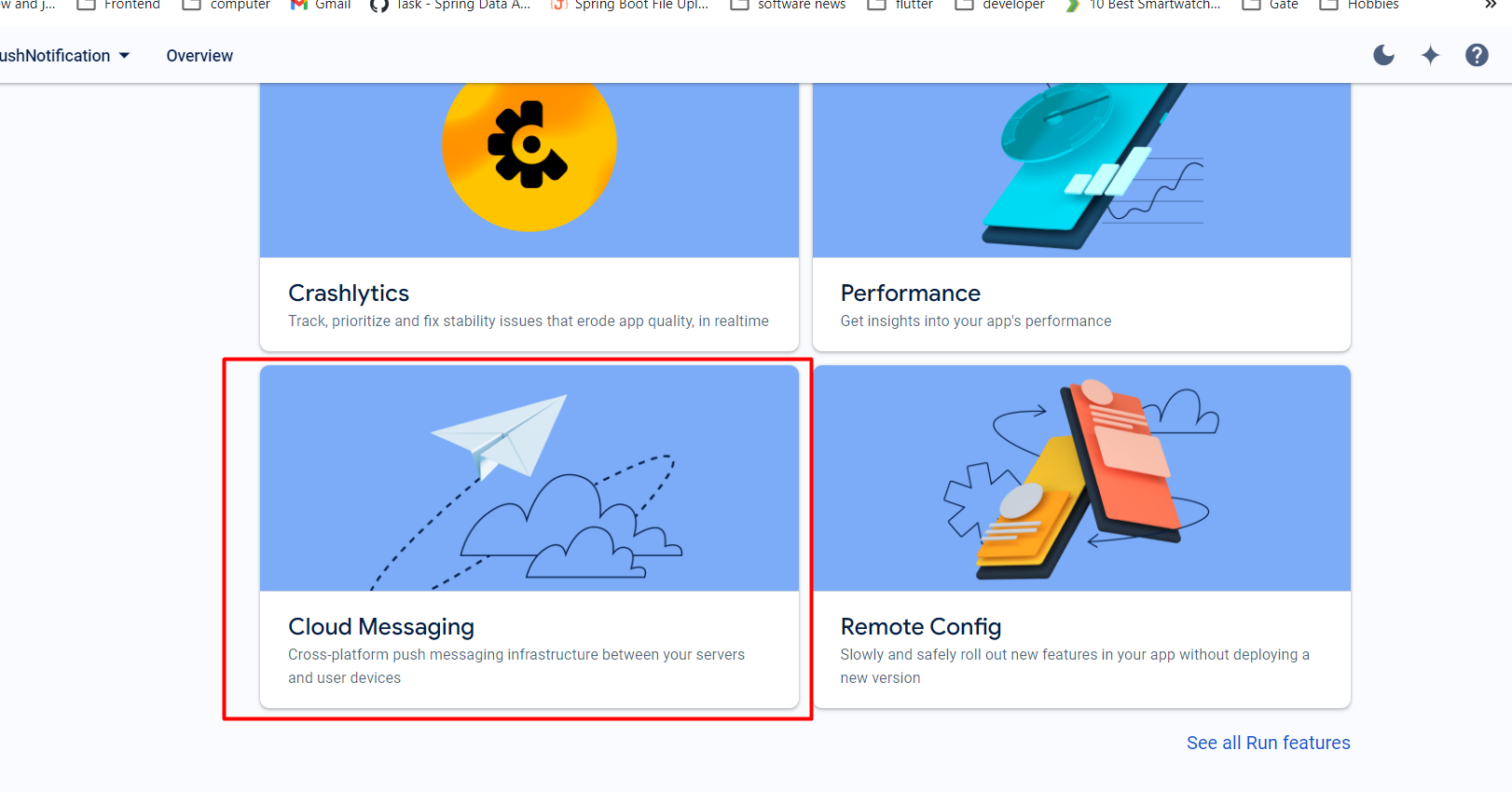
* Step 4: Disabled google analytics



* Project is created:



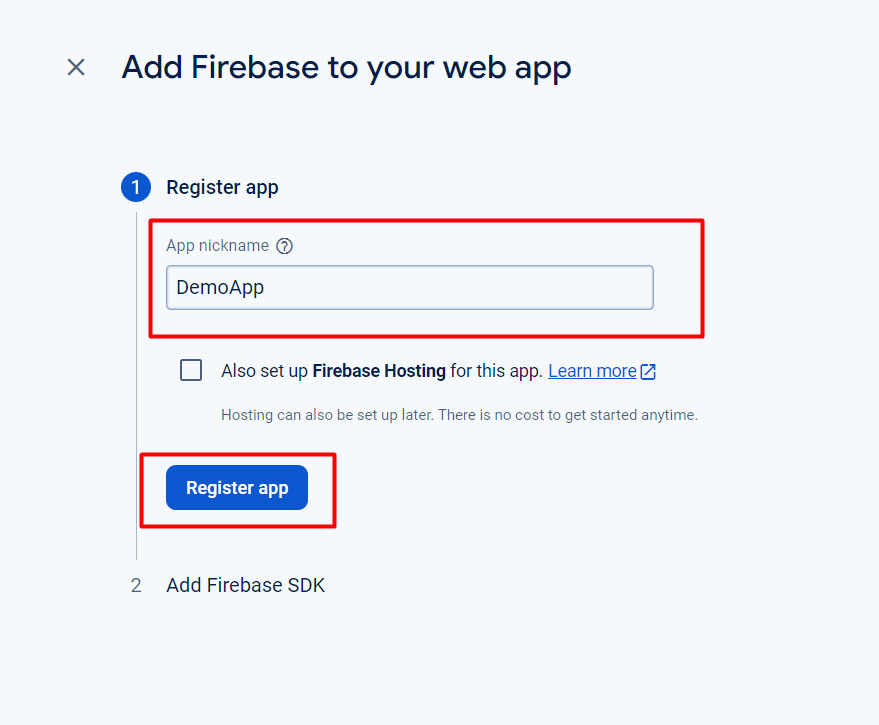
* Now scroll down and click on cloud messaging:

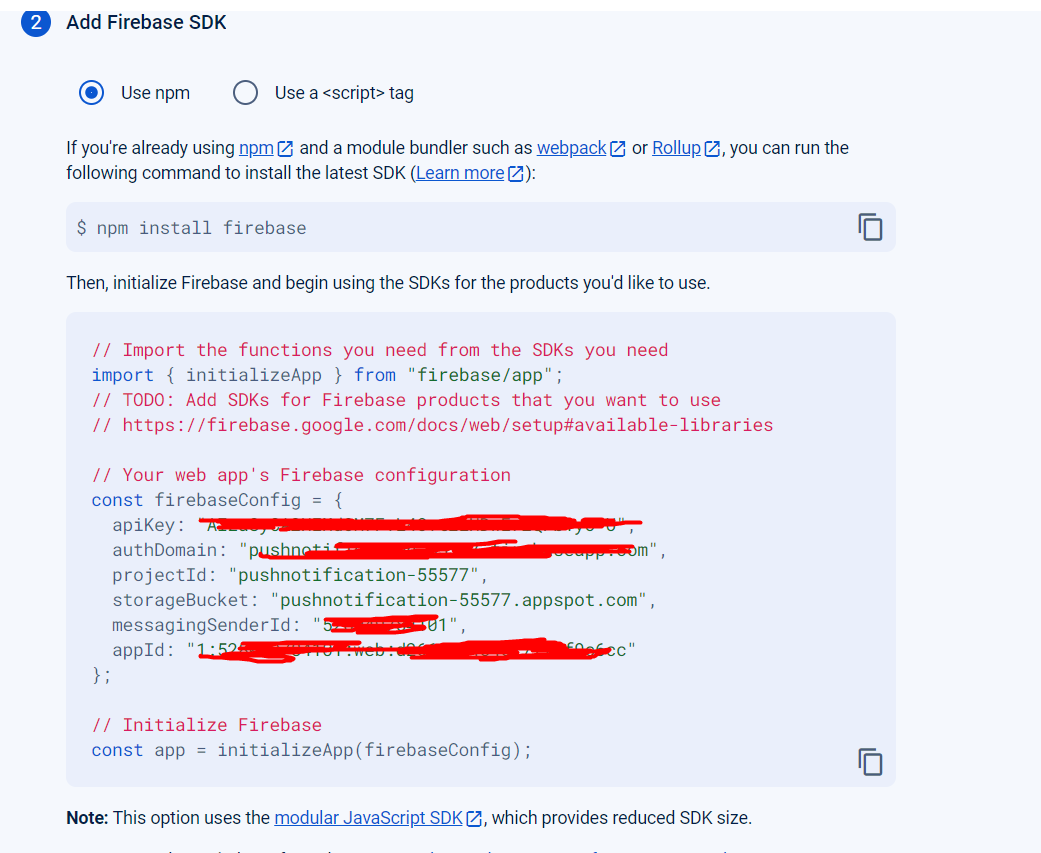
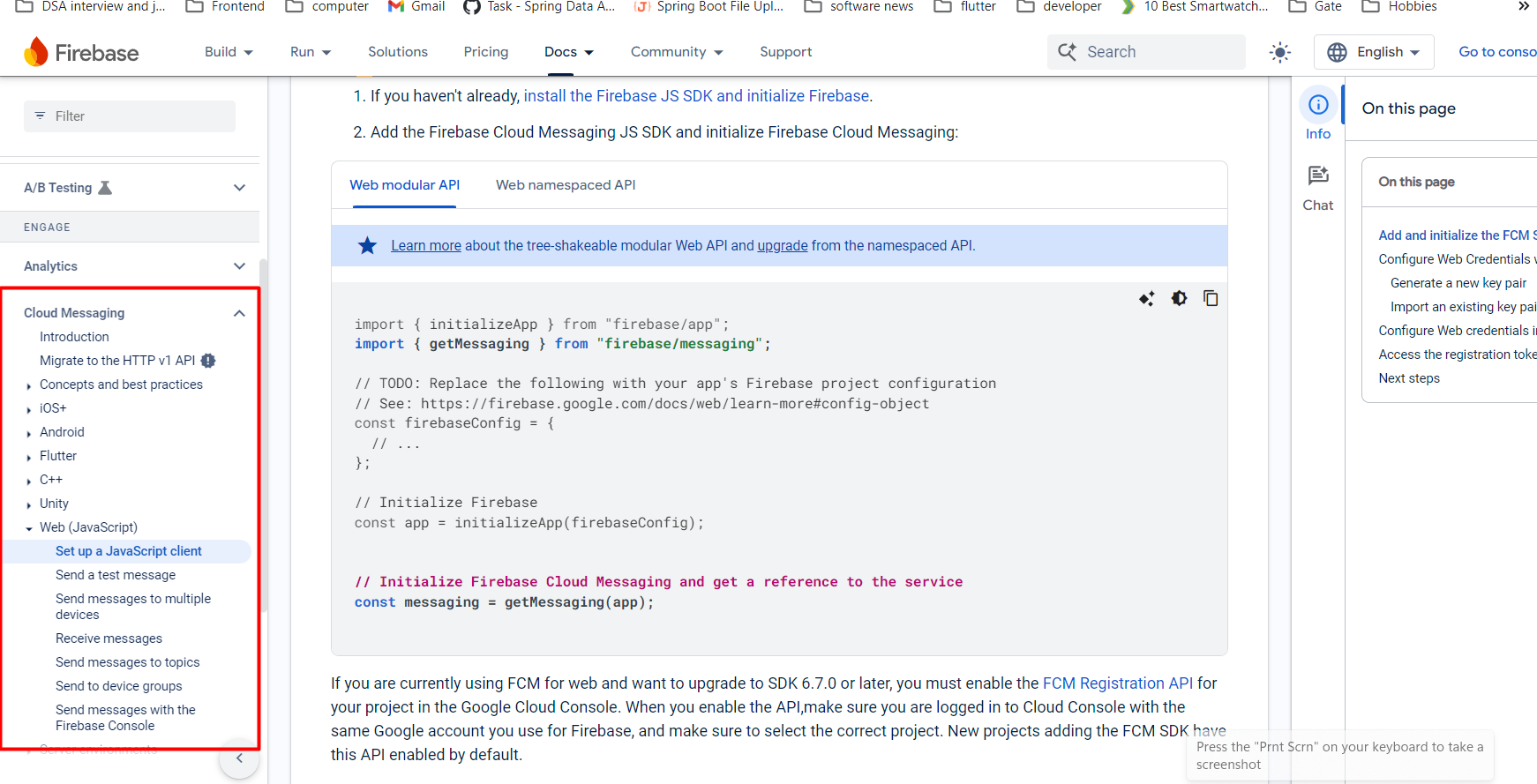
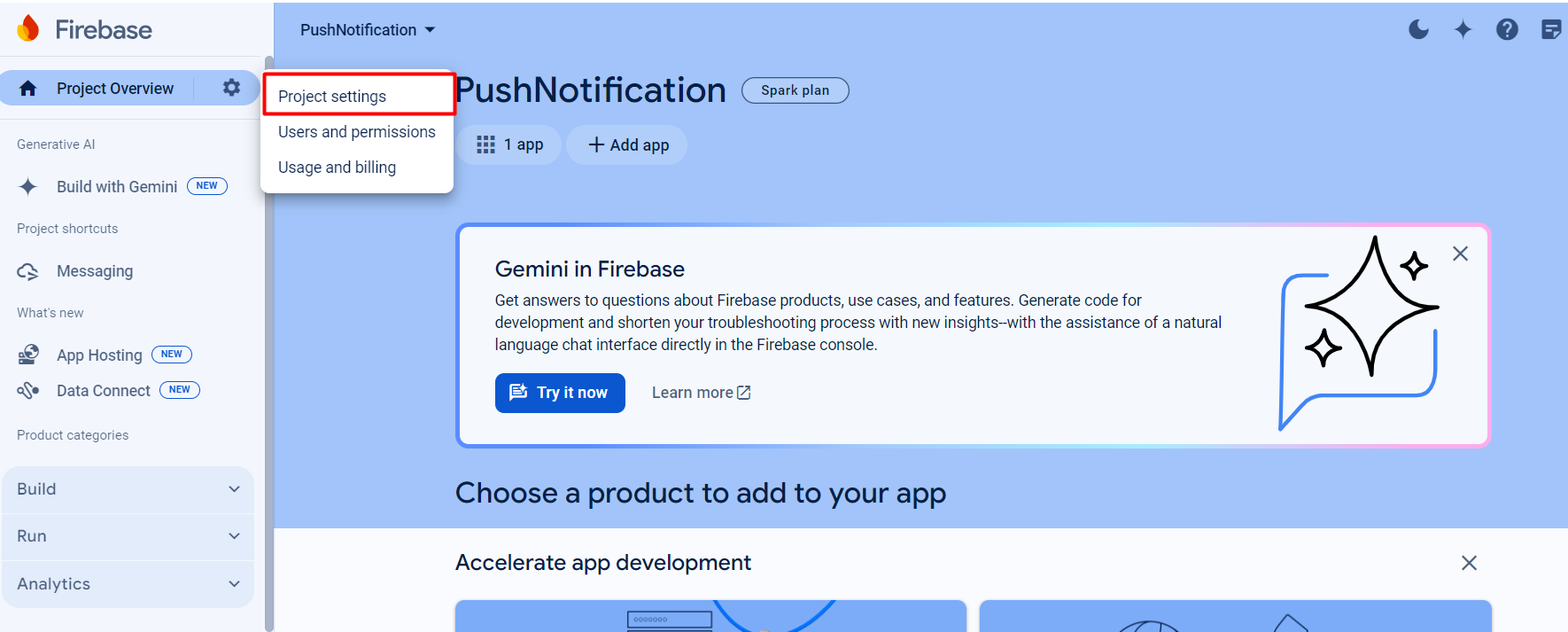
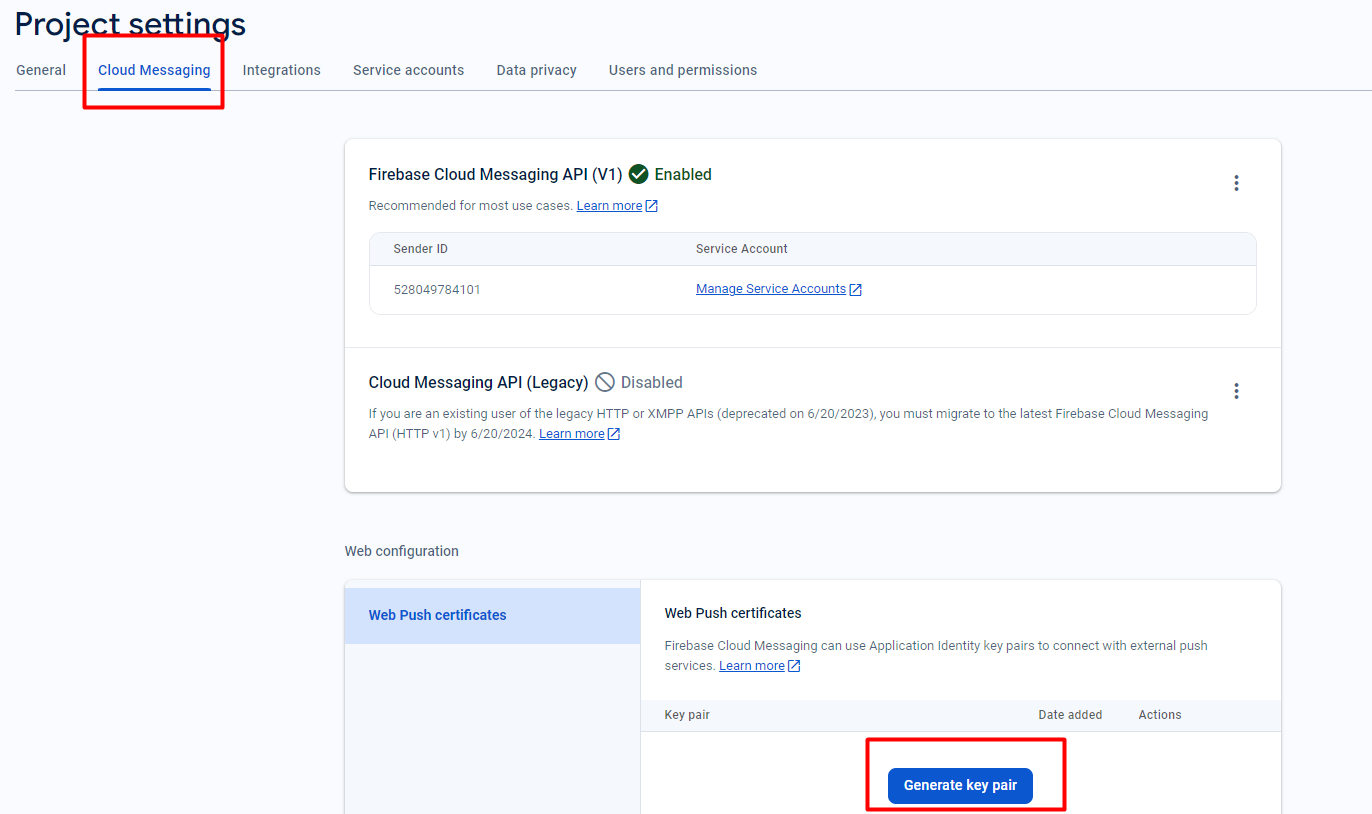
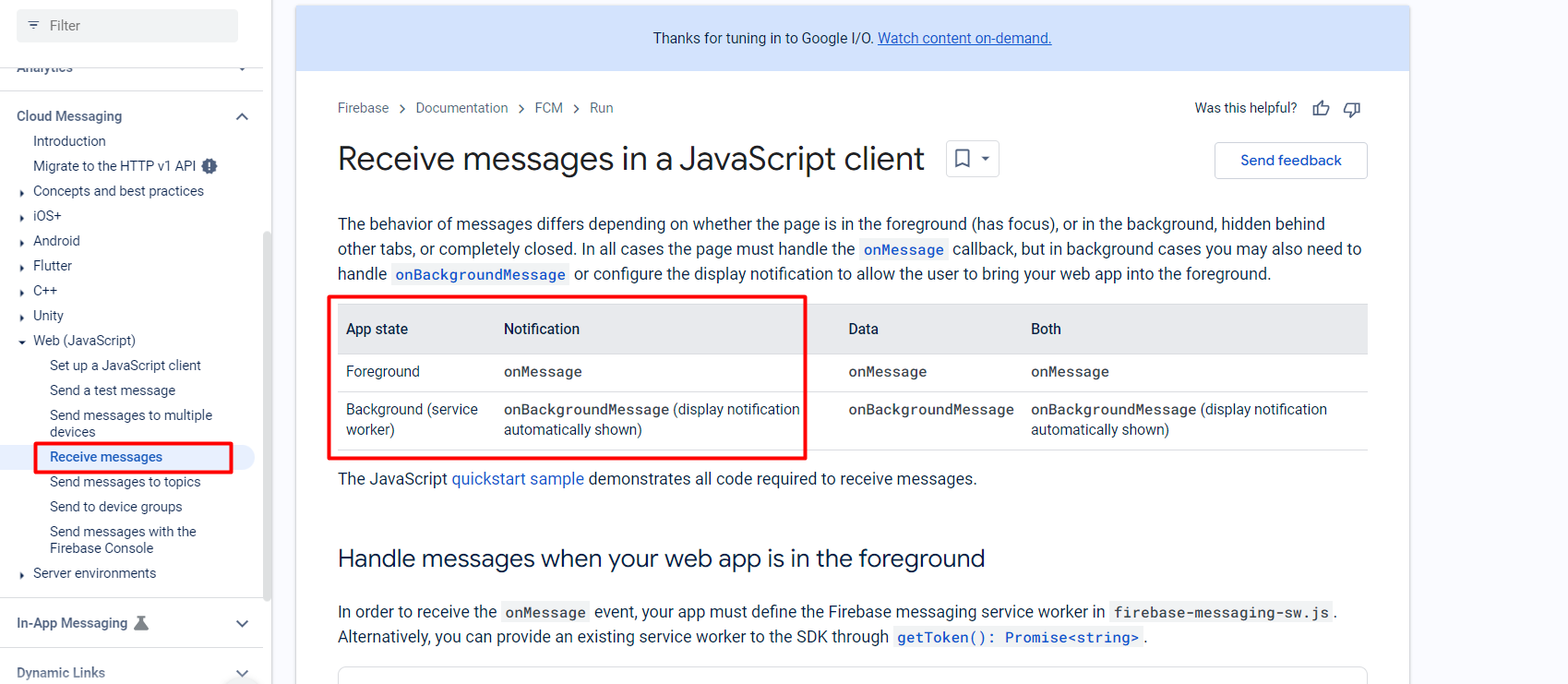
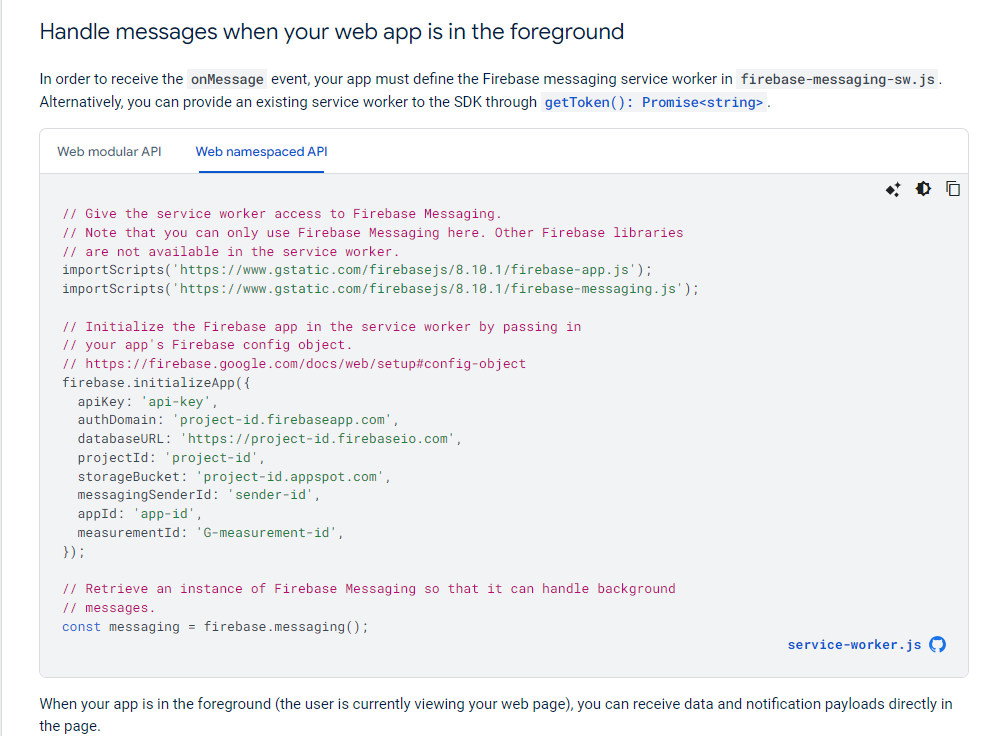
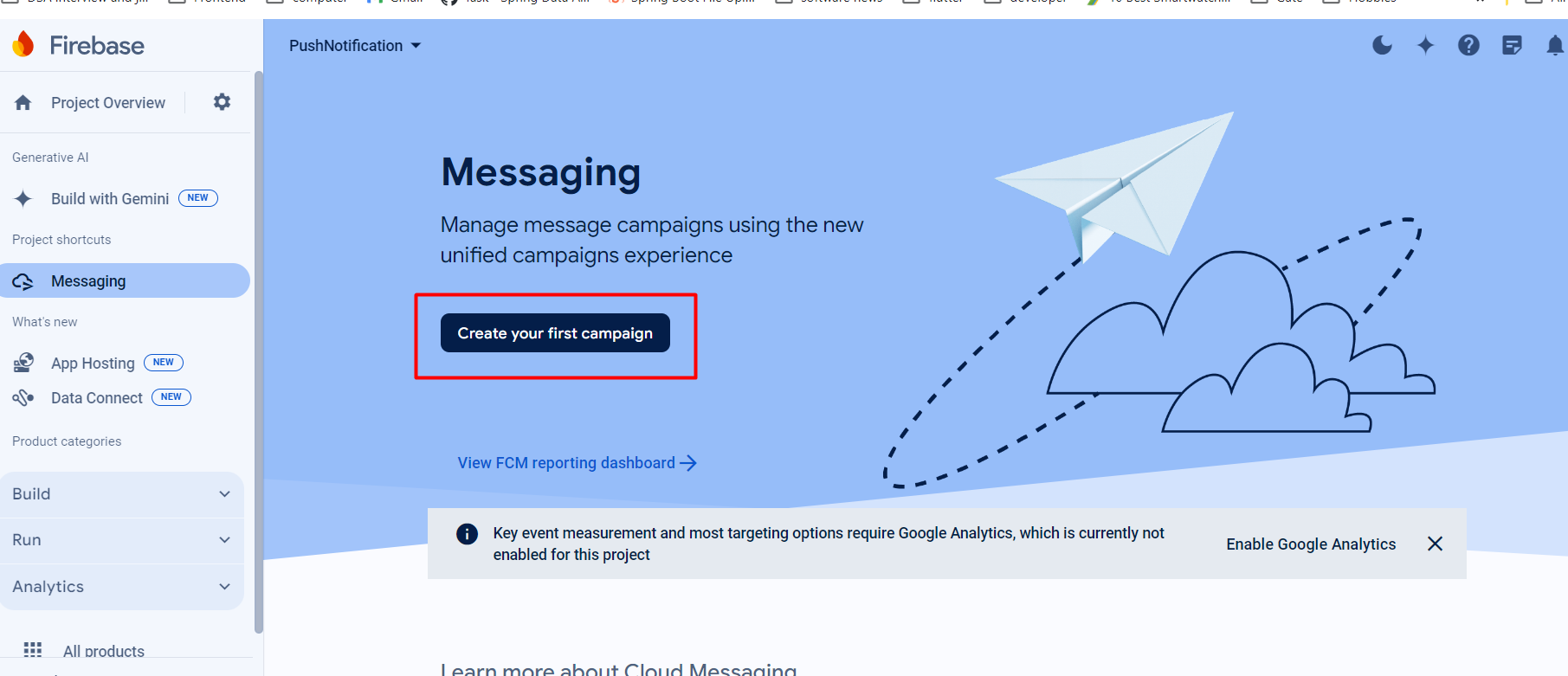
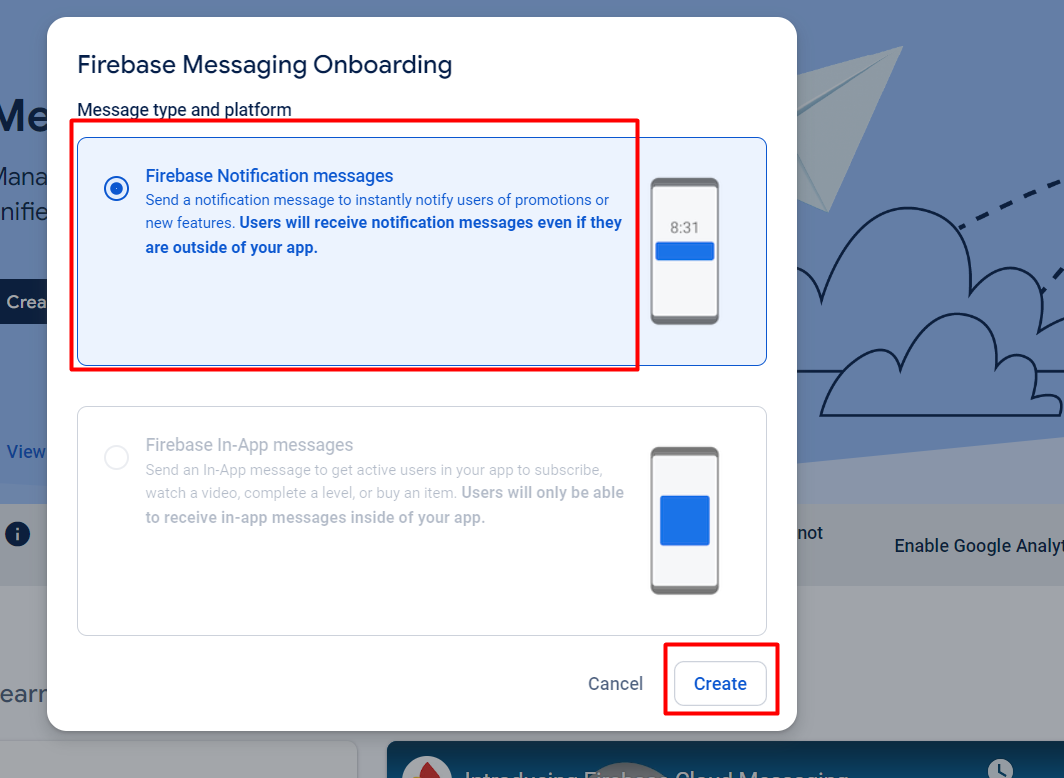
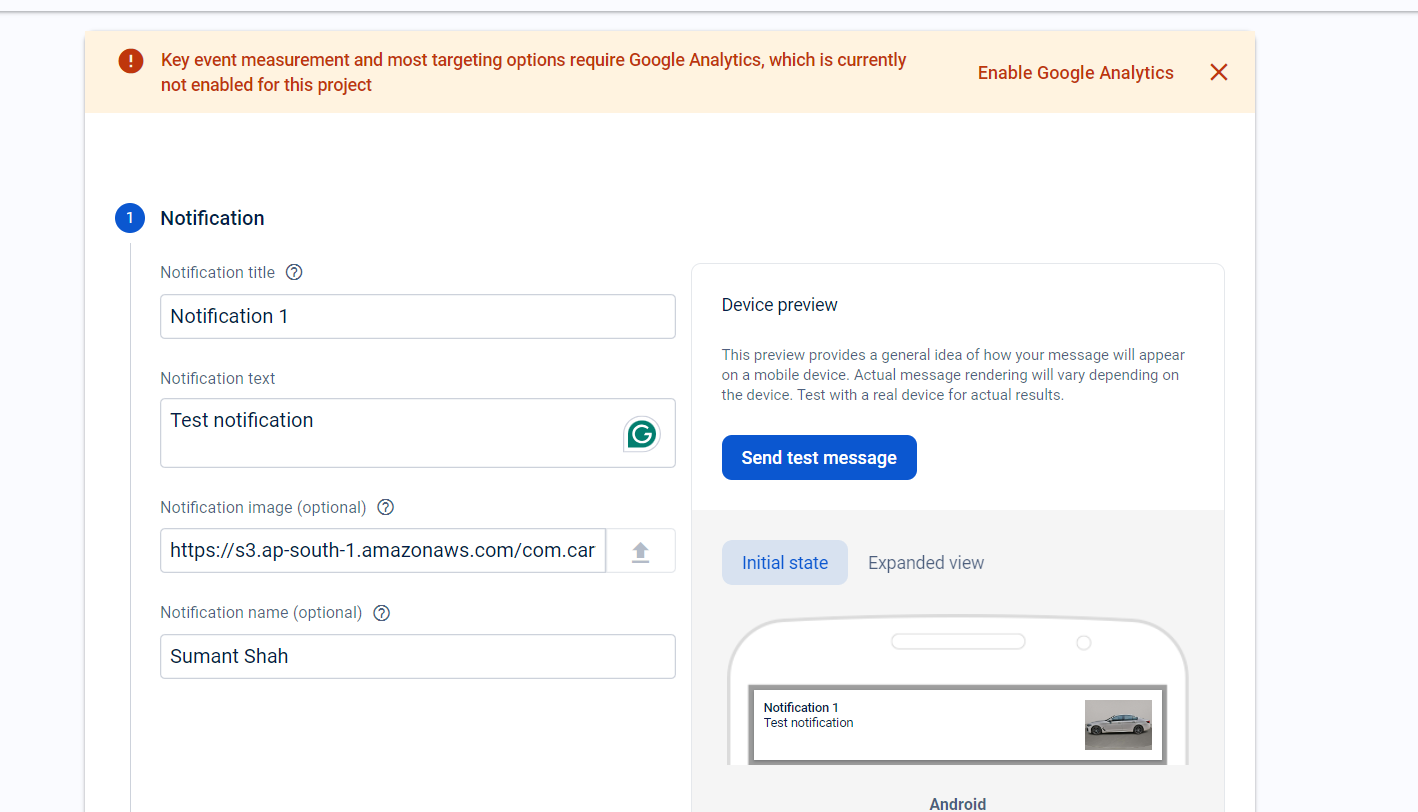
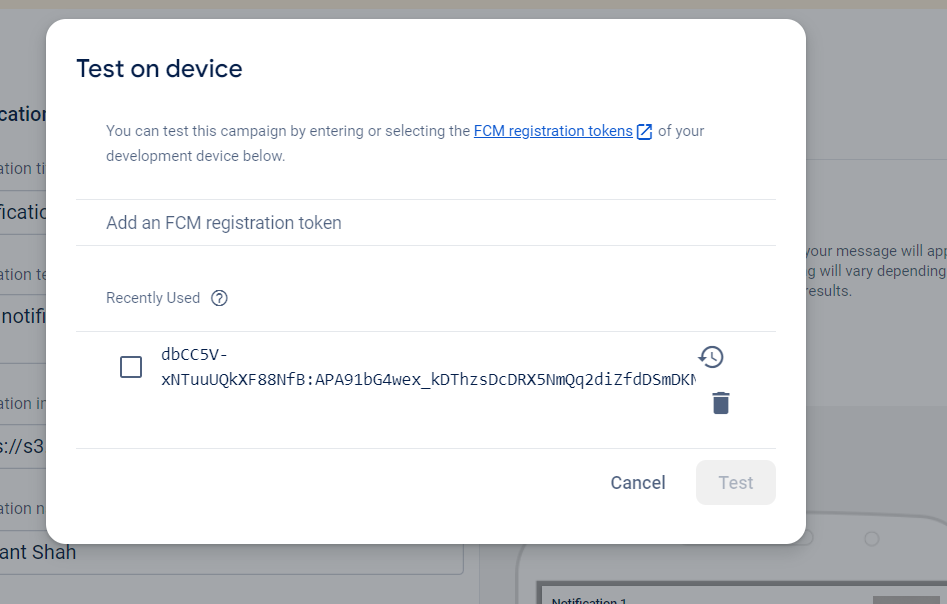
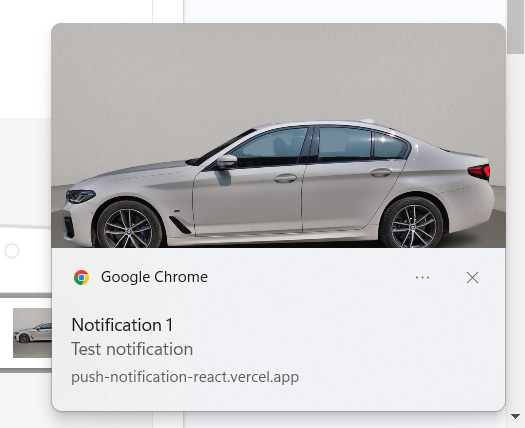
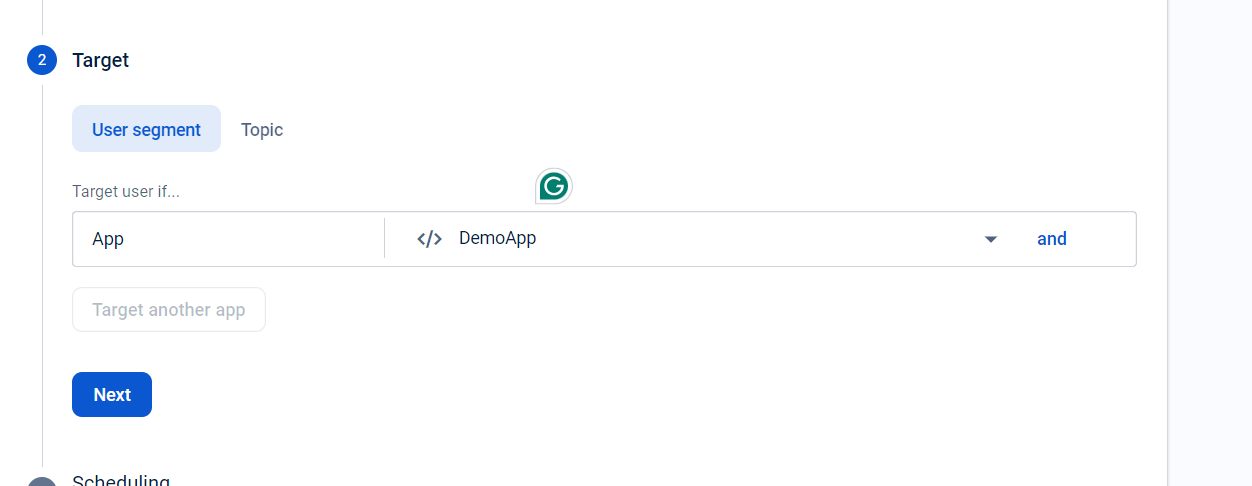
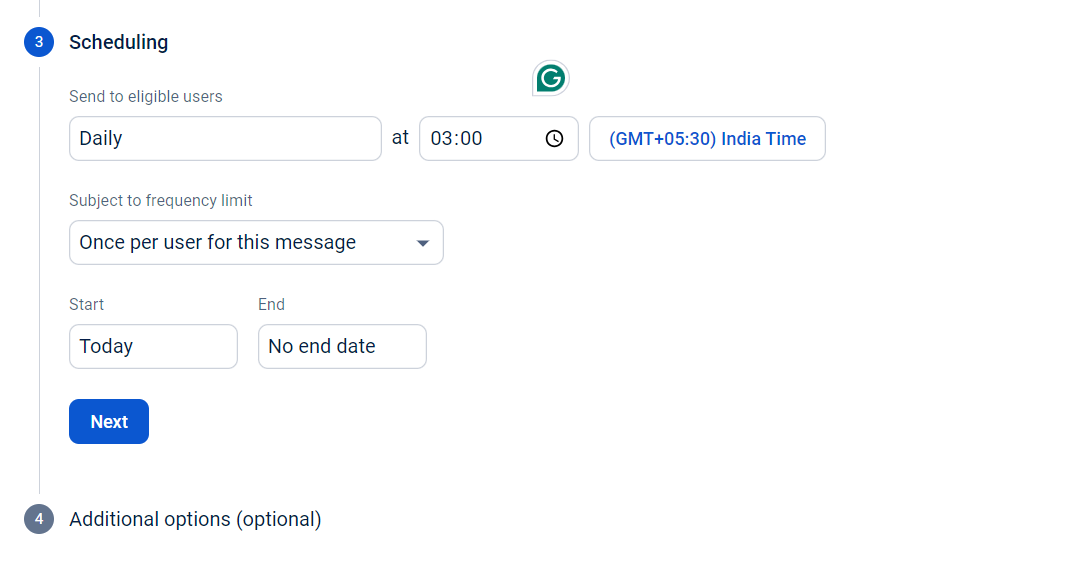


* Click on the web app logo:

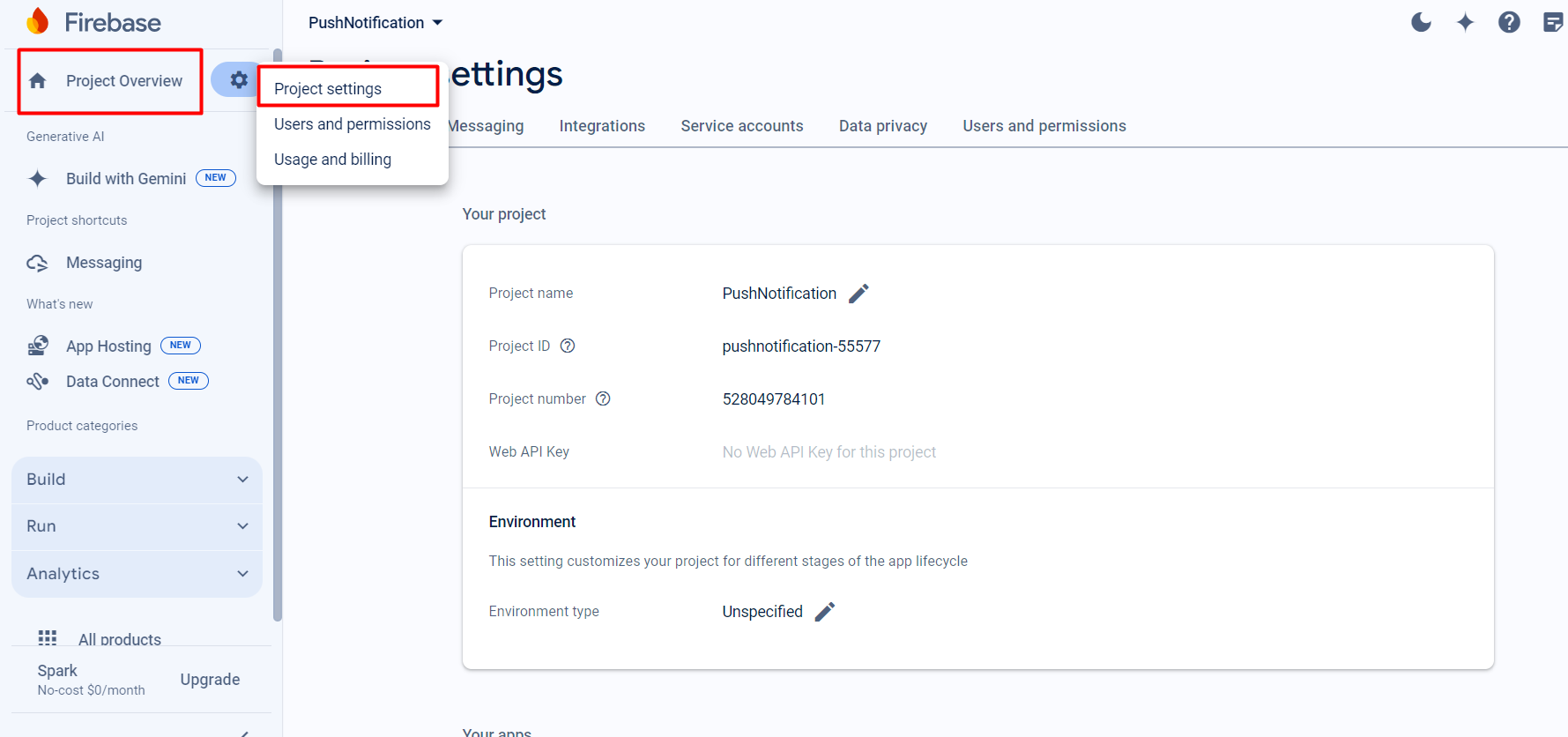
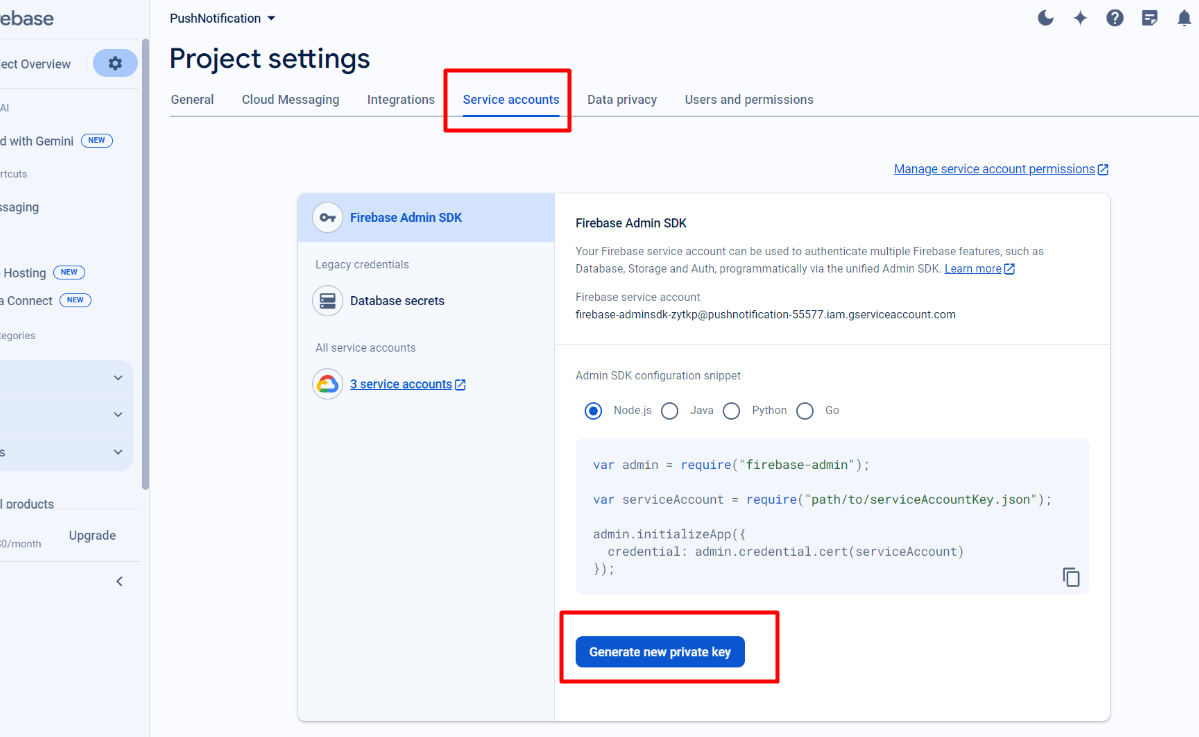


* Name the app and click on register app

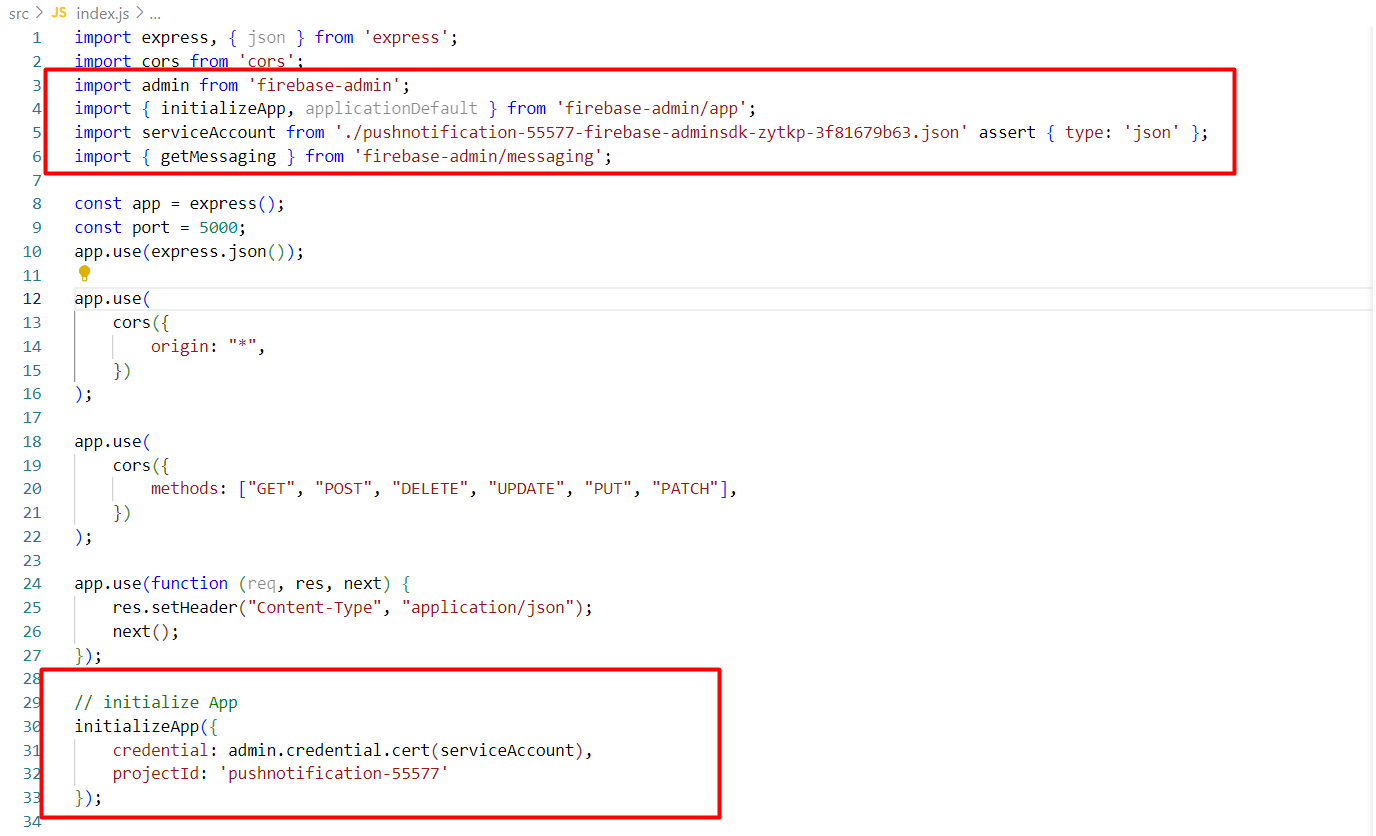
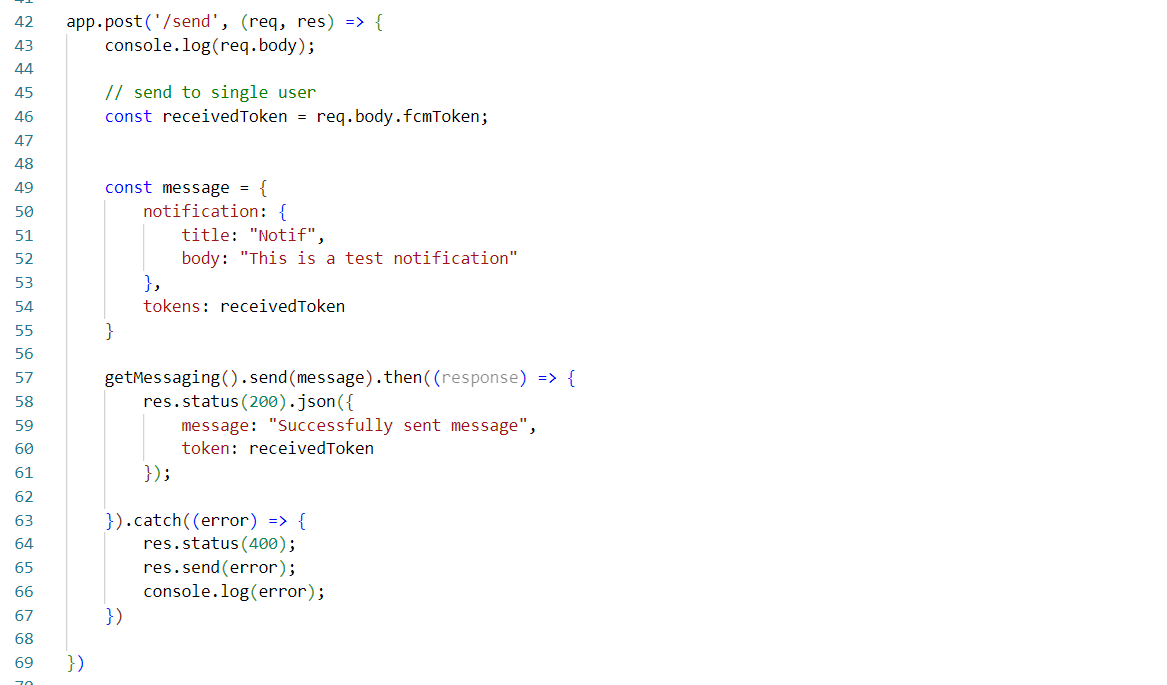


* Now the screen will display the integration we need to do in react app  
    
  After that click on finish and go to the console
* Then, open the google docs for the build and go to the cloud messaging section:  
    
  getMessaging thing we need to add in the code
* Then we need to go to Project settings of our created project  
  
* Go to cloud messaging and click on generate key pair  
    
  The key which is generated is called ‘vapid’ key and it will be used to generate the unique token.
* After making the function to generate token in the react app and giving the ‘vapid key’ to the react project, we need to work on receive messages from the doc  
  
  + Foreground means the app will be active then the notification will be handled by the app
  + Background messaging will be handled by the service worker.
* Next step is to integrate following in our code:  
  
* Then in the same file as above we need to add:  
  
* After creating everything, go to our project, click messaging from sidebar and click on “Create our first campaign”  
  
* next step  
  
* Then write the msg and icon to send and send a test message:  
  
  + You need to add the registration token:  
    
  + sample notification: 
* Next step:  
  
* next step:  
  
* Backend trigger with the device token:  
  

**Backend Setup (Node JS) for the Push Notification**

* In the project overview open project settings:  
  
* In the “Service Account” , generate new private key which will download the json file of credentials which we need to save in the backend project file  
  
* Then we need to create new node js project
* Dependencies installed in the project:  
  
* Paste the generated key file in the node js project folder.
* Do the basic setup of the project to start the server:

|  |
| --- |
| import express, { json } from 'express';  import cors from 'cors';  const app = express();  const port = 5000;  app.use(express.json());  app.use(      cors({          origin: "\*",      })  );  app.use(      cors({          methods: ["GET", "POST", "DELETE", "UPDATE", "PUT", "PATCH"],      })  );  app.use(function (req, res, next) {      res.setHeader("Content-Type", "application/json");      next();  });  app.get('/', (req, res) => {      res.send({ message: "hello" })  })  app.listen(port, () => {      console.log('Server is up on port ' + port);  }) |
|  |

* Connect the project with the firebase:
  + 
* The api for the notification sent to single user:
  + 
* The Api for the notification sent to multiple users:  
  