



Complete web Setup Guide(Frontend + BackEnd)

Thank You for your purchase.....

Thank you for choosing our code. We appreciate your purchase and aim to provide exceptional service. If you need help with the code or documentation, contact our [Support Team](#).

What you'll need

Server Requirement Notice

❖ VPS

- First you have to buy VPS from hosting providers like Hostinger, AWS ec2, etc, or any dedicated Server (or any hosting environment that supports [Node.js](#)).

⚠ Shared hosting environments (like cPanel or basic web hosting) are not supported, as they do not provide the **necessary permissions** to run [Node.js](#) applications properly.

- You have to buy an Ubuntu server.

❖ Root User Permissions

- You will need sudo user permission on the server to automate the project setup.

❖ Versions Requirements

- PHP [8.3](#)(minimum)
- Node.js (v[18](#) or higher)
- Need Composer version [2](#)
- [Root](#) or [SSH](#) access

Environment Configuration

❖ SSH into server :

- Then after you have to connect to server using SSH
- Navigate to the path where your zip file is located.
if directory **/var/html** not available then please create a folder

```
cd /var/html
```

- To unzip the file you have to install unzip.

```
sudo apt install unzip
```

- unzip the .zip file.

```
unzip your.zip
```

Domain Configuration

- ❖ Go to the DNS settings for your domain (on your domain registrar or hosting provider, e.g. GoDaddy, Namecheap, Cloudflare, etc.).

Type	Name	Value (Your Server IP)	TTL
A	@	YOUR.SERVER.IP	Auto or 3600
A	admin	YOUR.SERVER.IP	Auto or 3600

Server Configuration

- ❖ Now open your terminal to execute the commands

```
sudo apt update
sudo apt install -y nginx
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
sudo npm install -g pm2
```

- ❖ For PHP Configurations

```
sudo apt install php8.4 php8.4-fpm php8.4-mysql -y
sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
sudo apt install php-cli unzip curl php-mbstring git -y
sudo systemctl start php8.4-fpm
sudo systemctl enable php8.4-fpm
sudo chown -R $USER:$USER /var/www/myapp
sudo chmod -R 755 /var/www/myapp
```

- ❖ For My SQL Installation

```
sudo apt install mysql-server -y
sudo systemctl status mysql
sudo systemctl start mysql
sudo systemctl enable mysql
```

- ❖ For Creating database and user, **replace demo credentials with your credentials**

```
sudo mysql
```

```
CREATE DATABASE myapp_db;  
CREATE USER 'myapp_user'@'localhost' IDENTIFIED BY  
'StrongPassword123!';  
GRANT ALL PRIVILEGES ON myapp_db.* TO  
'myapp_user'@'localhost';  
FLUSH PRIVILEGES;  
exit;
```

❖ Configure Nginx

To configure the Nginx run the following command

```
sudo nano /etc/nginx/sites-available/web
```

This above Command Will create a file name web now copy-paste the below code snippet to that file

Code Snippet

```

server {
    listen 80;
    server_name yourdomain.app www.yourdomain.app;

    location / {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

server {
    listen 80;
    server_name admin.yourdomain.app;

    root /var/www/html/public;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.4-fpm.sock; # ensure php8.4-fpm matches
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}

```

❖ Enable site and restart Nginx:

Run these commands through your ide Terminal

```

sudo ln -s /etc/nginx/sites-available/web /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
sudo systemctl enable nginx

```

❖ Install SSL with Certbot

```
sudo apt install -y certbot python3-certbot-nginx
sudo certbot --nginx -d yourdomain.app -d www.yourdomain.app
sudo certbot --nginx -d admin.yourdomain.app
```

❖ Allow firewall

```
sudo ufw allow 80
sudo ufw allow 443
sudo ufw reload
```

❖ For storage permissions

```
sudo apt install acl -y
sudo setfacl -R -m u:www-data:rwx,g:www-data:rwx /var/www/html
sudo setfacl -dR -m u:www-data:rwx,g:www-data:rwx /var/www/html
chmod -R 775 storage
chmod -R 775 bootstrap/cache
```

❖ **Configure Background Processes**

Cron Job (for Scheduled Tasks)

Add this line to your server crontab:

```
crontab -e
* * * * * cd /path/to/your-project && php artisan schedule:run >> /dev/null 2>&1
```

Note: Replace `/path/to/your-project` with your actual project path.

Supervisor (for Queue Workers)

```
sudo apt install supervisor -y
sudo nano /etc/supervisor/conf.d/laravel-worker.conf
```

Below code need to paste in laravel-worker.conf

```
[program:laravel-worker]
process_name=%(program_name)s_%(process_num)02d
command=php /path/to/your-project/artisan queue:work --sleep=3 --tries=3
autostart=true
autorestart=true
user=www-data
numprocs=8
redirect_stderr=true
stdout_logfile=/path/to/your-project/storage/logs/worker.log
```

After above run these command

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start laravel-worker:*
```

❖ Setup Caching System (Recommended)

For performance improvements, use Redis as cache and queue driver.

```
sudo apt install redis-server -y
sudo systemctl start redis
sudo systemctl enable redis
sudo systemctl status redis
```

1. **Install Redis on your server.**
2. **Update .env:**

```
CACHE_DRIVER=redis
QUEUE_CONNECTION=redis
```

❖ EsimTel Server Settings

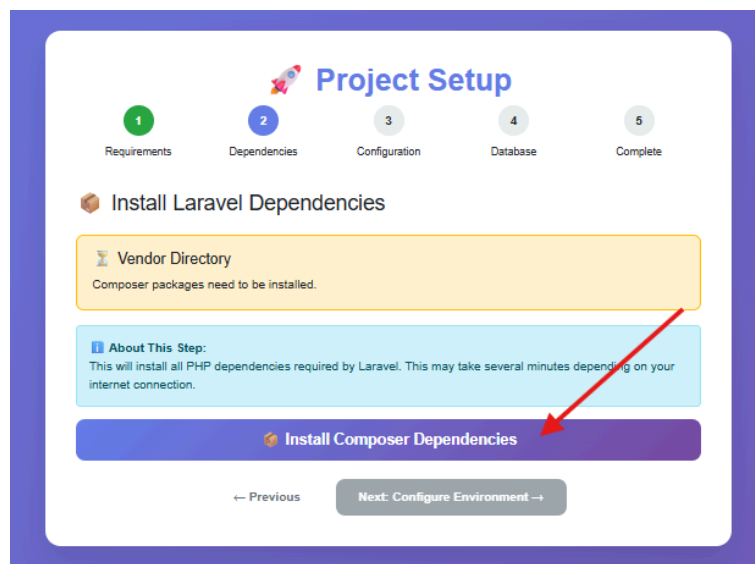
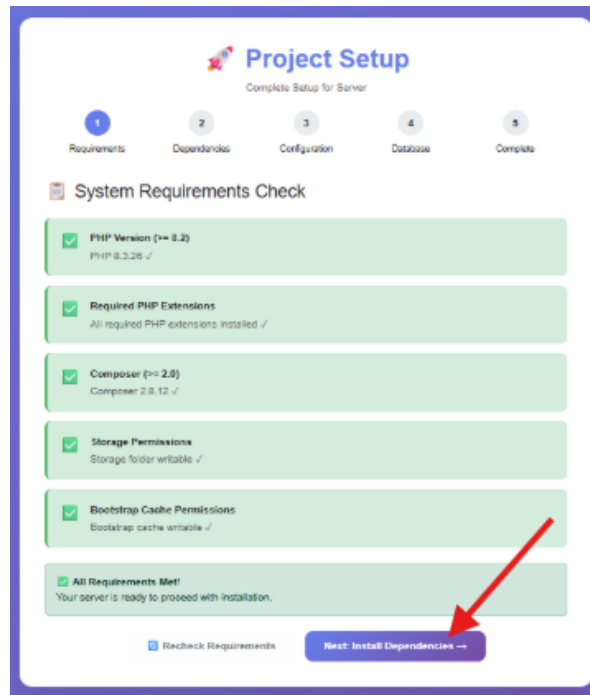
1. Remove Disabled Function in server (passthru, shell_exec, system, exec)

Back-end Configuration

Note: Make sure Laravel is running from the following directory:
`/var/www/html/your-laravel/public`

❖ System Requirements

Here we begin the back-end setup click on Next install Dependencies



If everything goes well, you will see a success message like the one shown in the screenshot below.

Note: If you encounter an **error**, try running the command through the terminal for **troubleshooting**.

1

Requirements

2

Dependencies

3

Configuration

4

Database

5

Complete

Project Setup

Environment Configuration

About This Step:
Configure basic application settings. An APP_KEY will be generated automatically for security.

Application Name

Laravel

Environment

Production

Debug Mode

Disabled (Recommended for Production)

Application URL

https://clever-wilbur.20-197-2-238.plesk.page

Save Configuration

← Previous

Next: Database Setup →

1

Requirements

2

Dependencies

3


Configuration

4

Database

5

Complete

Project Setup

Environment Configuration

About This Step:

Configure basic application settings. An APP_KEY will be generated automatically for security.

Application Name

Laravel

Environment

Production

Debug Mode

Disabled (Recommended for Production)

Application URL

https://clever-wilbur.20-197-2-238.plesk.page

Save Configuration

← Previous

Next: Database Setup →

1

Requirements

2

Dependencies

3


Configuration

4

Database

5

Complete

Project Setup

Environment Configuration

Environment configuration saved successfully!

About This Step:

Configure basic application settings. An APP_KEY will be generated automatically for security.

Application Name

Laravel

Environment

Production

Debug Mode

Disabled (Recommended for Production)

Application URL

https://clever-wilbur.20-197-2-238.plesk.page

Environment is already configured. You can update settings or proceed to the next step.

Update Configuration

← Previous

Next: Database Setup →

1 Requirements

2 Dependencies

3 Configuration

4 Database

5 Complete

Project Setup

Database Configuration

Database connection successful!

Database Type: MySQL Host: 127.0.0.1

Port: 3306 Database Name: esim_test

Username: esim_test Password:

[Test Database Connection](#)

Create Admin User (Required)

Important: You must create an admin user to access the application.

Admin Name *
Admin

Admin Email *
admin@example.com

Admin Password *
Minimum 8 characters

Confirm Password *

[Run Migrations & Create Admin](#)

[← Previous](#) [Complete Installation →](#)

Note: You will now need to set your own email and password for the admin. If you receive an error message, enter the admin email and password, then click **"Run Migration & Create Admin"** again.

Note: If you encounter an **error**, try running the command through the terminal for **migrate database**.

```
php artisan migrate  
php artisan db:seed
```

❖ Administrative Setup

1. Open the Admin Panel Login Page in your browser.
2. Log in with the admin credentials you created during installation.
3. Go to General Settings and configure:
 - Upload your logo.
 - Add payment gateway keys (Stripe, Razorpay, etc.).
 - Enter your Airalo API keys for eSIM functionality.

❖ Configure SMTP Details

The **.env** file contains all sensitive project configurations(You can Set Here in **.env**)

Mail Setup

MAIL_HOST=smtp.gmail.com

MAIL_PORT=587

MAIL_USERNAME=your-email@example.com

MAIL_PASSWORD=your-app-password

MAIL_ENCRYPTION=tls

MAIL_FROM_ADDRESS=no-reply@example.com

Firebase & FCM Setup (Push Notifications)

FCM_SERVER_KEY=your-fcm-server-key

FIREBASE_CREDENTIALS=storage/app/google/firebase.json

➡ Upload your Firebase service account JSON file to:
storage/app/google/firebase.json

This is required for Firebase Storage and Push Notifications.

System Requirements for front-end

Steps to Run This Project Successfully

- Make **.env.local** file and add this value given below , use this for SEO

NEXT_PUBLIC_DOMAIN= yourdomain.example

- ❖ Now go to the terminal and run command "**npm install**" to fetch the necessary dependencies that we have used.
- ❖ To Start your development process run command "**npm run dev**" now the project runs successfully

How to Create the Build

- ❖ In order to create build please run the following command given below

"npm run build"

- ❖ Start via PM2

```
pm2 start npm --name "next-app" -- run start
pm2 save
pm2 startup
```

Other Configurations

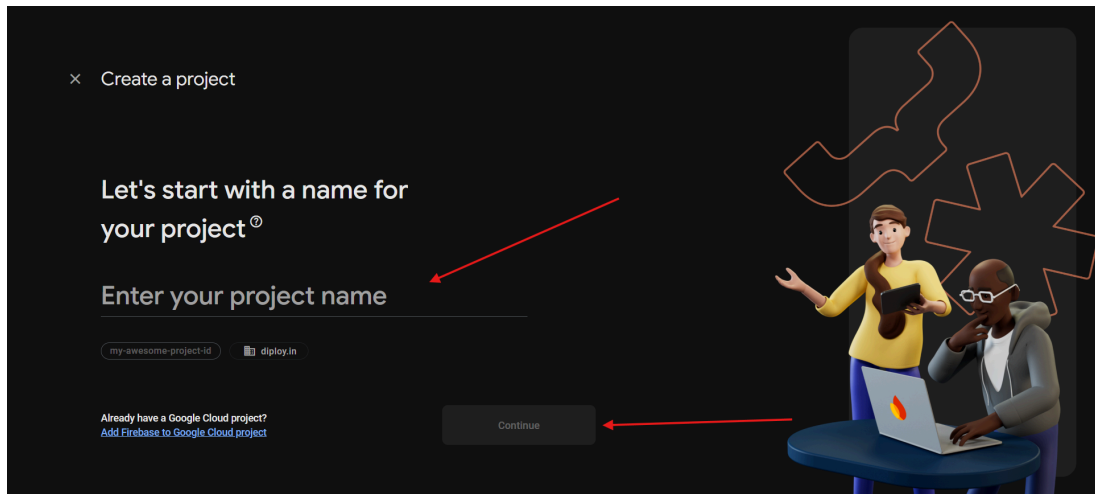
Lets see how you can change the language

- ❖ To add a new static word for a language, navigate to:

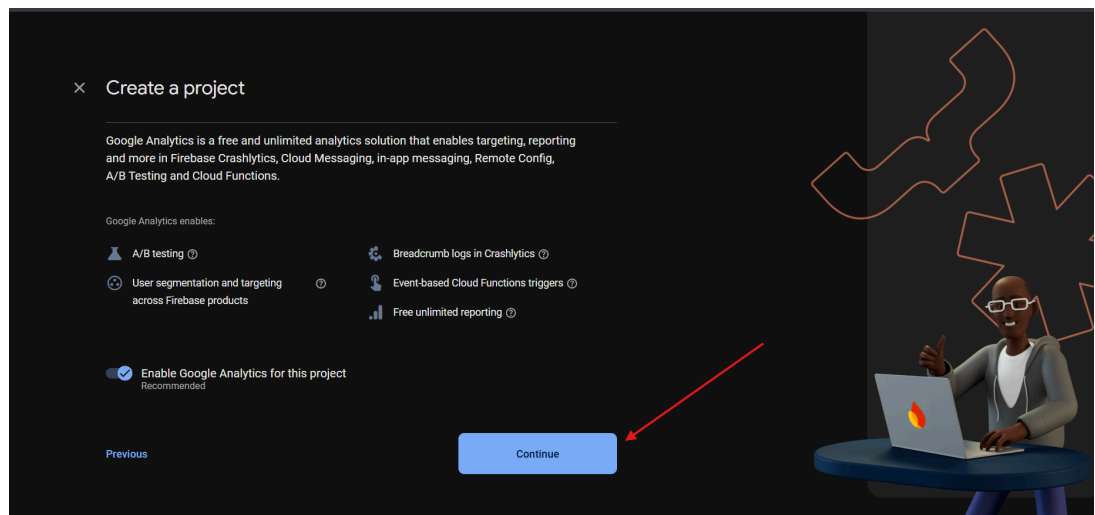
```
root/
├── src/
│   ├── locales/
│   │   ├── en.json/
│   │   └── *.json/
│   └── # Static assets
│       # for language
│       # create new file per language
```

Configure Your Firebase

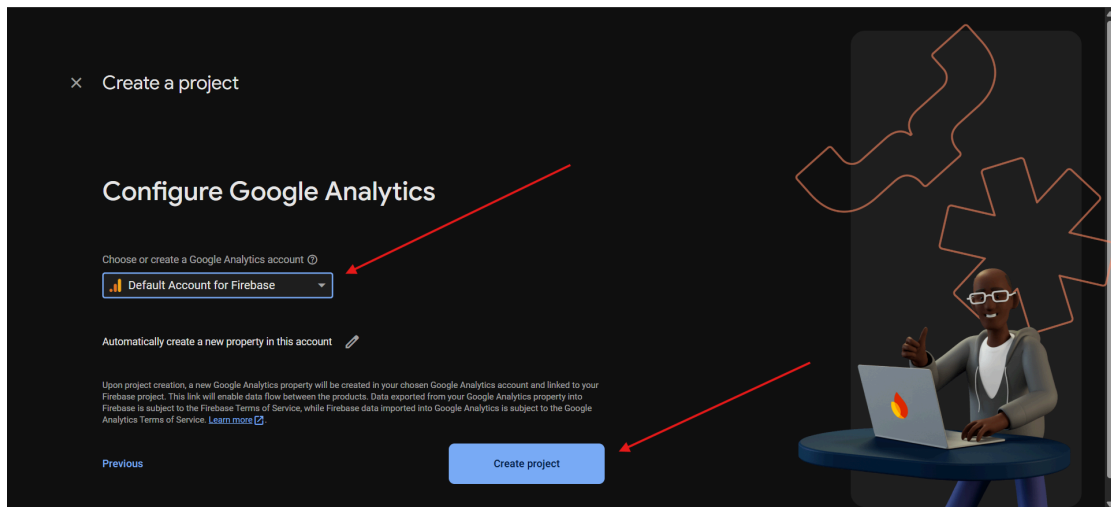
- ❖ Go to official firebase Website to click here [firebase](#)
- ❖ Click on create new project and Enter your project name here, now click on continue



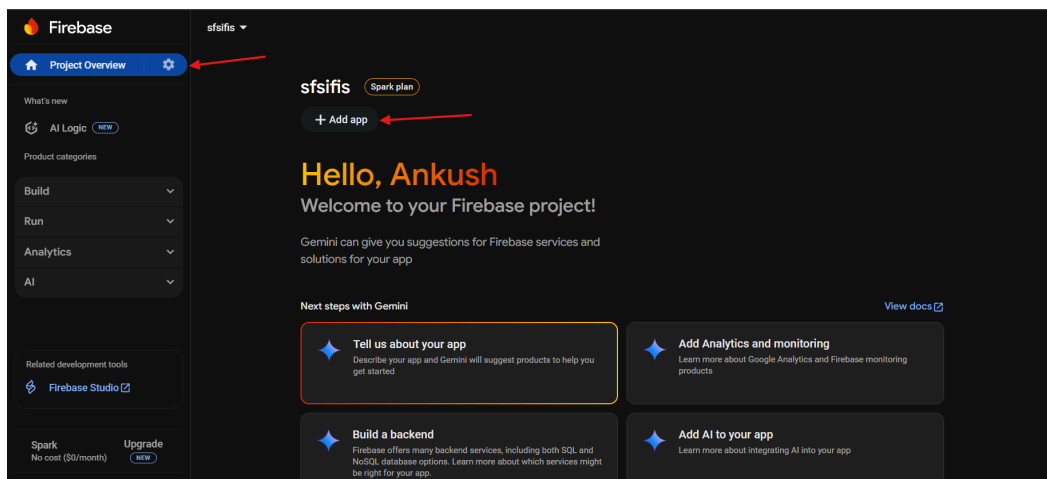
❖ Click on Continue



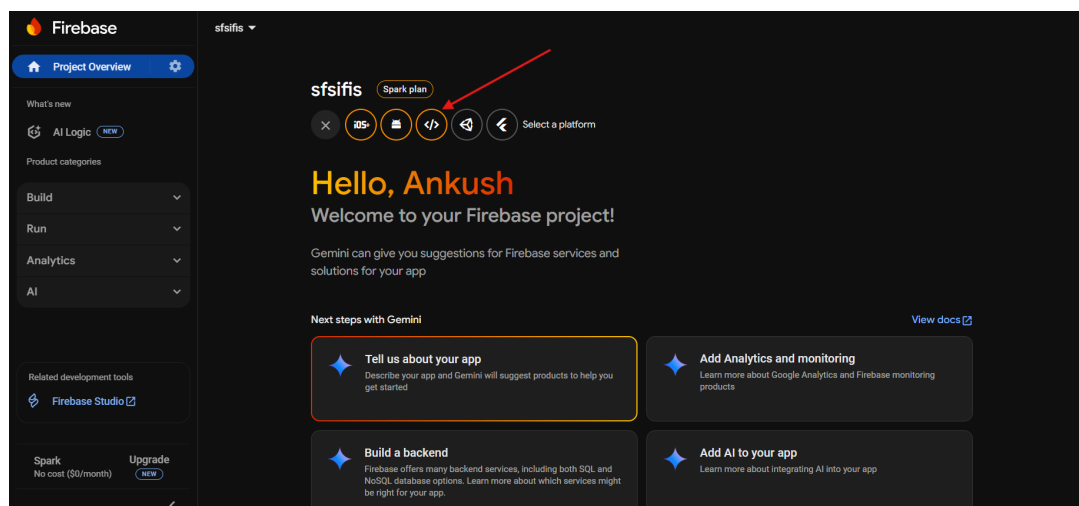
- ❖ Now Select the **Default Account for Firebase** and click on create project



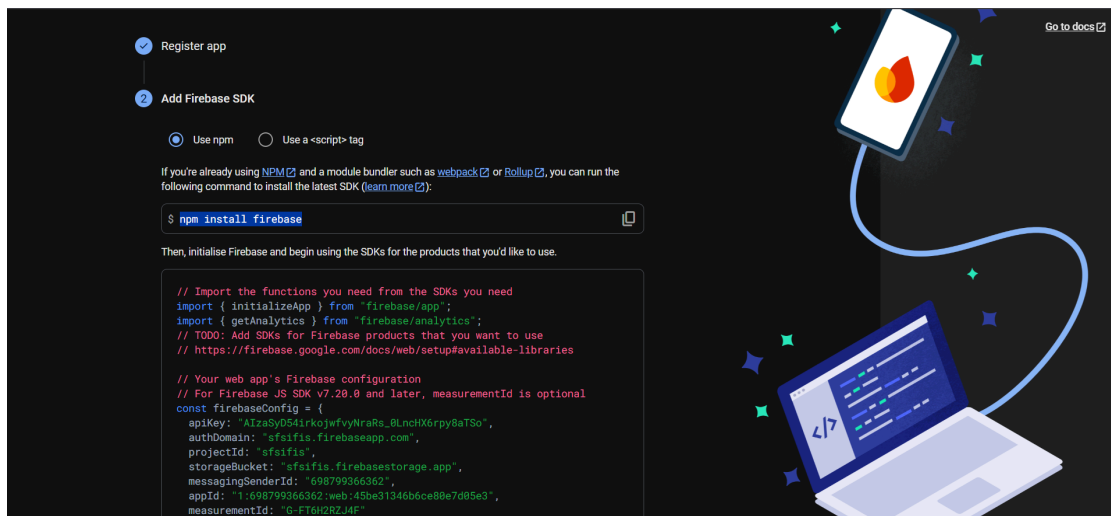
- ❖ Please wait for a minute until your project creates successfully, once the project is created the click on continue. Now go to the Project settings and click on add app



- ❖ Now Click on the Web Icon



- ❖ Now put the Project nick name and next then install the npm command given below and click on continue to console



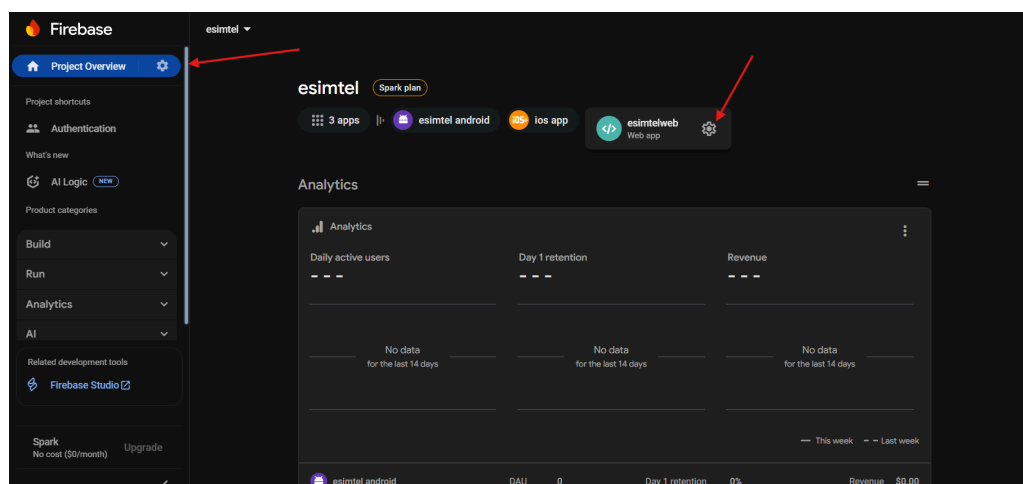
- ❖ Wooh 🏆 Now your firebase project was created successfully.

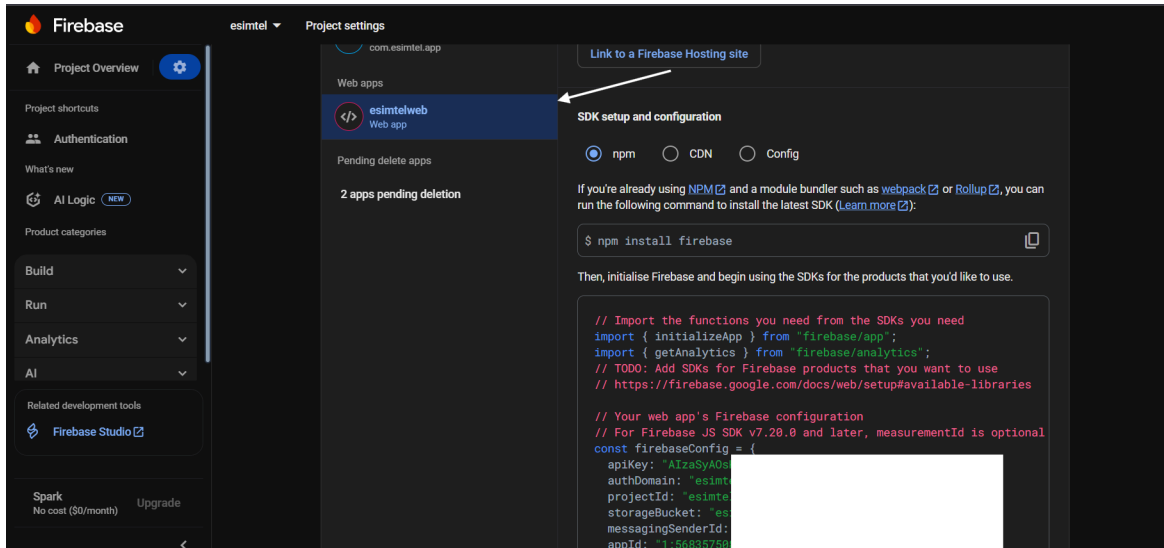
How To Connect The firebase app to the [Next.js](#) web application

For Firebase notification and Google login, add all credentials in Admin web settings

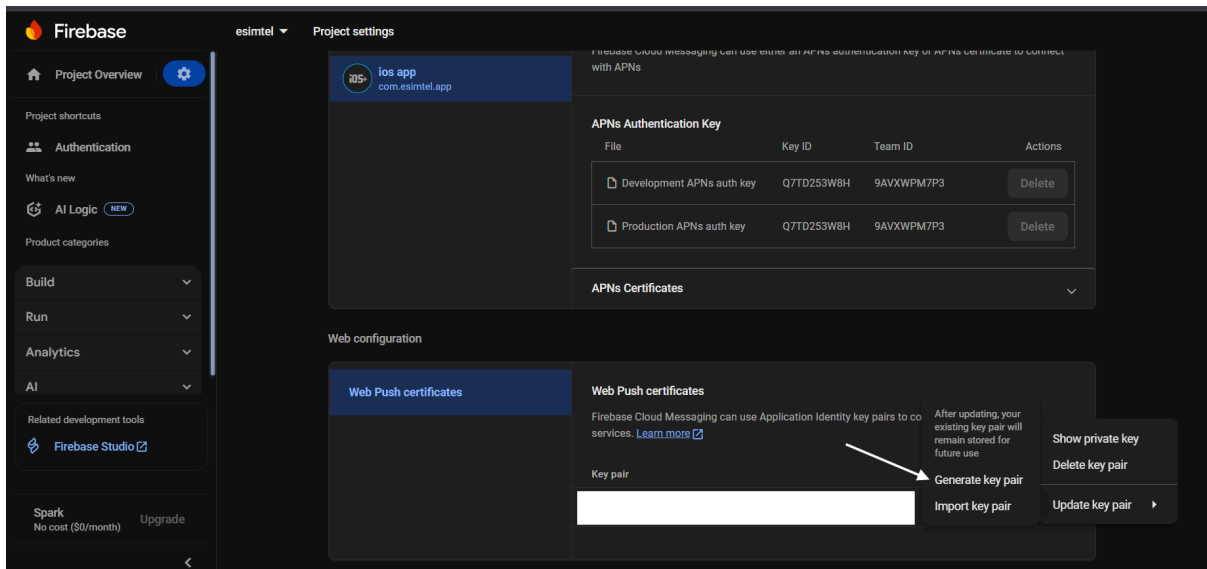
- FIREBASE_API_KEY="**your_api_key**"
- FIREBASE_AUTH_DOMAIN="**your_auth_domain**"
- FIREBASE_PROJECT_ID="**your_project_id**"
- FIREBASE_STORAGE_BUCKET="**your_storage_bucket**"
- FIREBASE_MESSAGING_SENDER_ID="**your_sender_id**"
- FIREBASE_APP_ID="**your_app_id**"
- FIREBASE_VAPID_KEY="**your_vapid_key**"

Note: You can file all the Keys in the Firebase console





You can create the **FIREBASE_VAPID_KEY** to go to project settings-> cloud messaging scroll down click on Generate key pair



Contact Support

Delighted to serve and support you! Contact our support team. We're available Mon-Fri, 10:00 am - 7:00 pm IST (GMT +5.30),

India - Asia. Expect prompt responses within 24 hours via comments, forum, or email.

Reach Us

Unlock the potential of diploy, your go-to hub for top-notch Flutter app development and web applications. Reach out to us today for quality solutions and exciting freelance opportunities.

Visit us at diploy.in or drop an email to start@diploy.in

Thank you for considering us!