

Astroway Admin Panel

Thank You for your purchase

Thank you for choosing our code. We appreciate your purchase and aim to provide exceptional service.

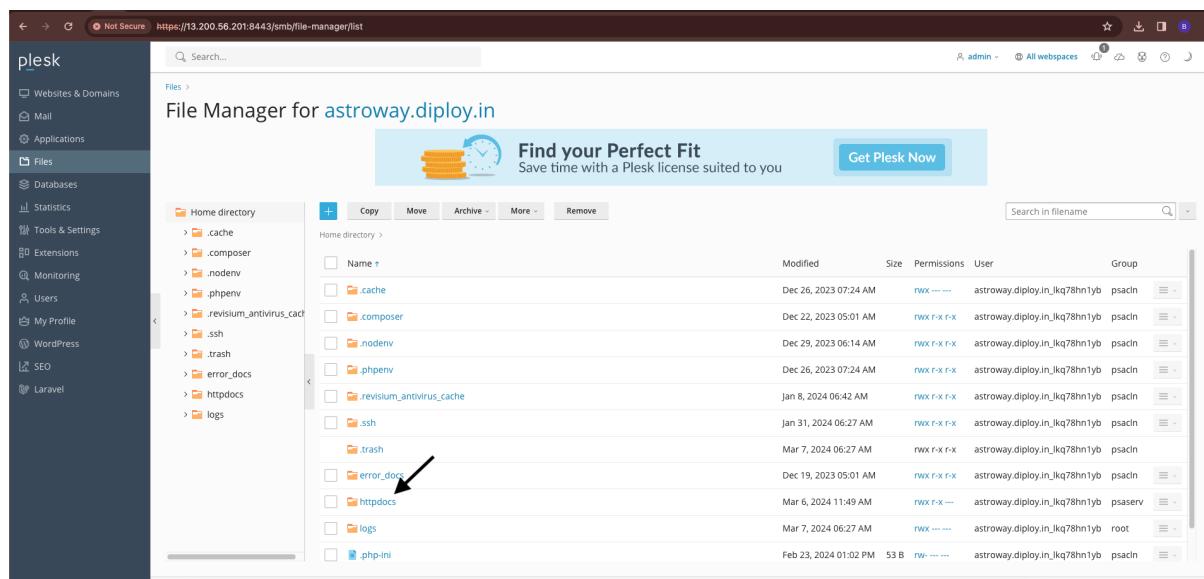
If you need help with the code or documentation, contact our support.

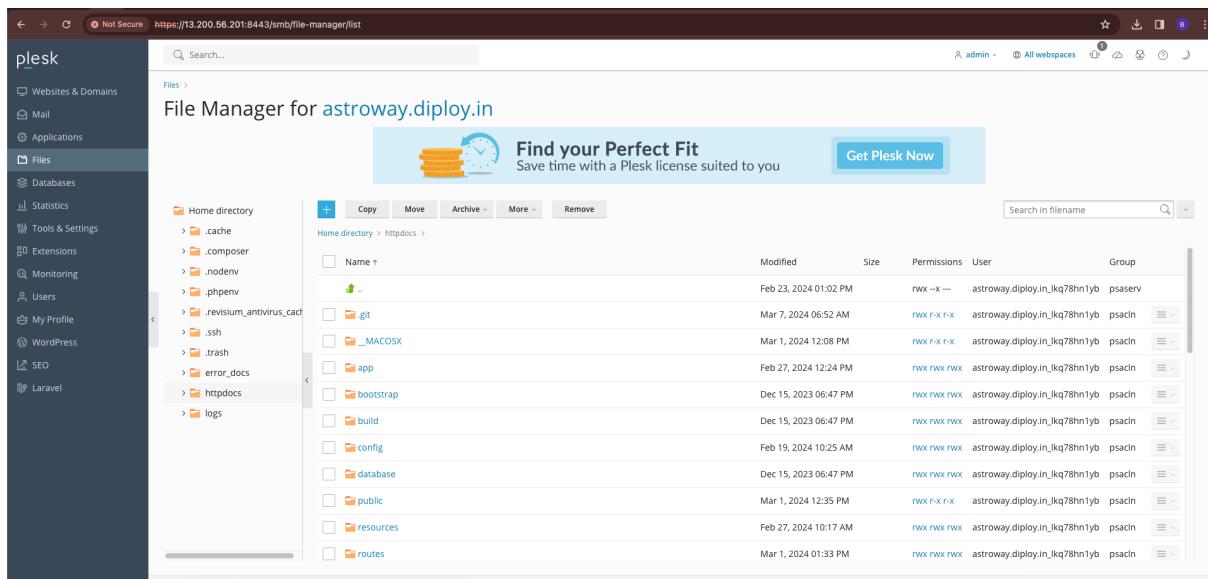
Setup Admin Panel.

- 1) To Setup Admin Panel First Of All You need To Upload This Project On Server.
- 2) It can be any server for example like hostinger,cpanel,plesk etc.
- 3) PHP version must be **8.2 or greater**

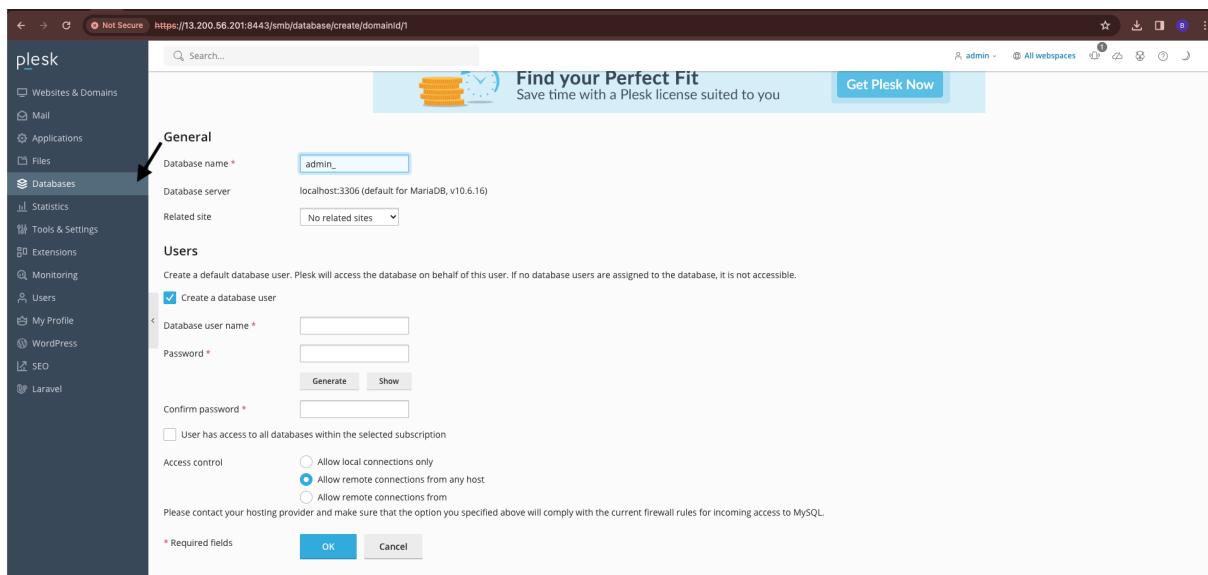
Upload Project On Server

- 1) For Example I am Going to Upload it on plesk You can use any of the server which i mentioned above
- 2) In Plesk Open httpdocs folder and upload your project. If you are using other server like hostinger or cpanel then you need to open the folder public_html folder and upload the project.

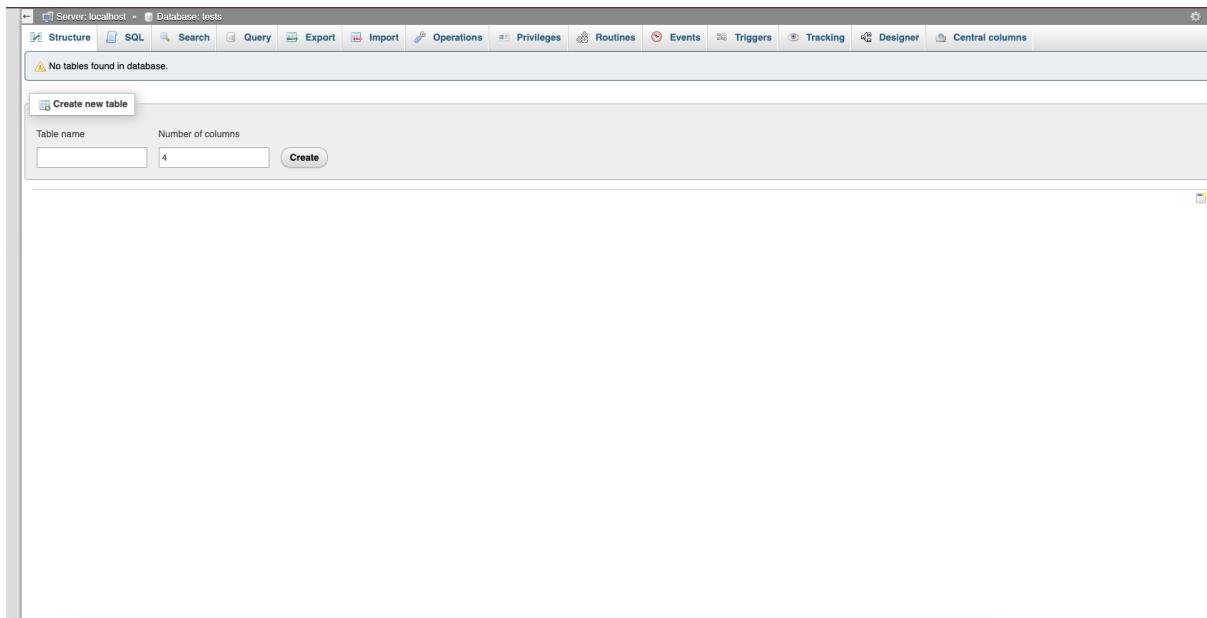




3) After that you need to upload sql file in database section.



4) Setup Database Name , username and password.After setup that open phpmyadmin In that upload the sql file using drag and drop.



5) Then You Need To Setup the .env file from your project and add database credentials in that which you created.

```

1 APP_NAME=Astroyay
2 APP_ENV=production
3 APP_KEY=base64:rSH/odv19zYFVigar/8j0g1LgBz5f0feAJDtBLoJ%=
4 APP_DEBUG=true
5 APP_URL=https://localhost
6
7
8 LOG_CHANNEL=stack
9 LOG_DEPRECATIONS_CHANNEL=null
10 LOG_LEVEL=debug
11
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=db_astroway
15 DB_USERNAME=root
16 DB_PASSWORD=
17 DB_PASSWORD=
18
19 BROADCAST_DRIVER=log
20 CACHE_DRIVER=file
21 FILESYSTEM_DISK=local
22 QUEUE_CONNECTION=sync
23 SESSION_DRIVER=file
24 SESSION_LIFETIME=120
25
26 MEMCACHED_HOST=127.0.0.1
27
28 REDIS_HOST=127.0.0.1
29 REDIS_PASSWORD=null
30 REDIS_PORT=6379
31
32 MAIL_DRIVER=smtp
33 MAIL_HOST=mailtrap
34 MAIL_PORT=587
35 MAIL_ENCRYPTION=null
36 MAIL_PASSWORD=null
37 MAIL_ENCRYPTION=null
38 MAIL_FROM_ADDRESS="Hello@example.com"
39 MAIL_FROM_NAME="${APP_NAME}"
40
41 AWS_ACCESS_KEY_ID=
42 AWS_SECRET_ACCESS_KEY=
43 AWS_DEFAULT_REGION=us-east-1
44 AWS_BUCKET=
45 AWS_USE_PATH_STYLE_ENDPOINT=false
46
47 PUSHER_APP_ID=

```

DB_DATABASE= Database name

DB_USERNAME =Database user name

DB_PASSWORD= Database Password

And After that your database will successfully configured.

.env Setup

- 1) After Setup of Database.
- 2) APP_NAME = your app name
- 3) APP_URL = your domain url in which project you uploaded
- 4) Now There is below FCM_SERVER_KEY is there.

```

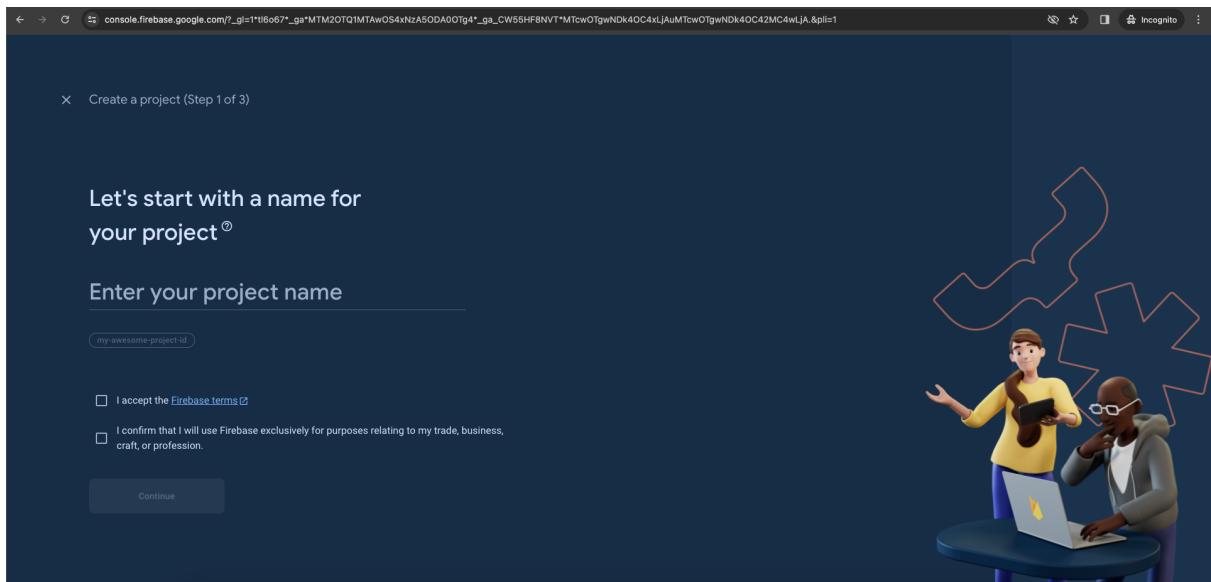
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_MAILER=smtplib
MAIL_HOST=tcp://127.0.0.1:587
MAIL_PORT=587
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="(APP NAME)"
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false
PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER=mt1
VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
VITE_PUSHER_HOST="${PUSHER_HOST}"
VITE_PUSHER_PORT="${PUSHER_PORT}"
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
JWT_SECRET=AQcRvNeze5FU28pPVzAPNzq9tSHAKLsZEisij5qYOK83bVYjDEHs1leCecT
FCM_SERVER_KEY = ████████████████████████████████████████████████████████████████████████████████

```

That Key Will Get From Firebase .Now Let get this from fire base

Firebase Setup

- First go to <https://console.firebaseio.google.com/> and create project.



- After Creating Project Just Go to Settings.In That Open Cloud Messaging

Firebase Project settings - Cloud Messaging

Firebase Cloud Messaging API (V1) Enabled

Cloud Messaging API (Legacy) Disabled

Web configuration

Service accounts

Sender ID	Service Account
[REDACTED]	Manage Service Accounts

3) Click on manage service account you will redirect to cloud

<https://console.cloud.google.com/>

Google Cloud - Astroways

Service accounts

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
firebase-adminsdk-98b7@astroways-befe4.iam.gserviceaccount.com	Enabled	firebase-adminsdk	Firebase Admin SDK Service Agent	No keys			[REDACTED]

Actions

- Manage details
- Manage permissions** [REDACTED]
- Manage keys
- View metrics
- View logs
- Disable
- Delete

4) In that click on manage keys .

The screenshot shows the Google Cloud IAM & Admin service account details page for 'firebase-adminsdk'. The 'KEYS' tab is selected. A red arrow points to the 'Create new key' button. The interface includes a sidebar with various IAM and Admin options like IAM, Identity & Organization, Policy Troubleshooter, and Service Accounts.

From there just create new keys

5) A popup will open in that select json and create your key

The screenshot shows a modal dialog titled 'Create private key for "firebase-adminsdk"'. It contains instructions about the private key and two radio buttons for 'Key type': 'JSON' (selected) and 'P12'. A red arrow points to the 'CREATE' button at the bottom right of the dialog. The background shows the same Google Cloud IAM & Admin service account details page as the previous screenshot.

6) Now You Will get the key add this key to your .env file



The screenshot shows the VS Code interface with the Explorer sidebar open, displaying a folder structure for a project named 'HTDOCS'. The '.env' file is selected in the center editor area. A red arrow points to the line 'FOI_SERVER_KEY = ' in the code, which is followed by a long string of characters. The entire string is highlighted with a red bar, indicating it has been redacted.

```
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_MAILER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=<hello@example.com>
MAIL_FROM_NAME="${APP_NAME}"
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false
PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER=mt1
VITE_PUSHER_APP_KEY="$(PUSHER_APP_KEY)"
VITE_PUSHER_HOST="$(PUSHER_HOST)"
VITE_PUSHER_PORT="$(PUSHER_PORT)"
VITE_PUSHER_SCHEME="$(PUSHER_SCHEME)"
VITE_PUSHER_APP_CLUSTER="$(PUSHER_APP_CLUSTER)"
JWT_SECRET=>0cRWez5fU2ppPvZAFWz0tSHAKL5ZEzij5gYOKB3BvYjDEHc1leCzeCt
FOI_SERVER_KEY = 
```

7) Now Open the json which is downloaded from firebase and copy the whole code

```
(1) astroways-bef4-7927272bf522.json x
Users > apple > Downloads > (1) astroways-bef4-7927272bf522.json > ...
1  {
2    "type": "redacted",
3    "project_id": "redacted",
4    "private_key_id": "redacted",
5    "private_key": "redacted",
6    "client_email": "redacted",
7    "client_id": "redacted",
8    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9    "token_uri": "https://oauth2.googleapis.com/token", Follow link (cmd + click)
10   "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11   "client_x509_cert_url": "redacted",
12   "universe_domain": "googleapis.com"
13 }
14
```

8) Now in your project open app/services/FCMService.php

The screenshot shows a developer's environment with a code editor and a browser. The code editor displays `FCMService.php` with several lines of PHP code. The browser tab shows a page titled "FCMService". The code includes a class definition for `FCMService` and a static method `send` that sends notifications to devices.

```
class FCMService {
    // Public property to store the service account key
    public $serviceAccountKey = [
        "type"=>"service_account",
        "private_key"=>[REDACTED],
        "private_key_id"=>[REDACTED],
        "private_key_md5"=>[REDACTED],
        "client_email"=>[REDACTED],
        "client_id"=>[REDACTED],
        "auth_uri"=>[REDACTED],
        "token_url"=>"https://oauth2.googleapis.com/token",
        "auth_provider_x509_cert_url"=>"https://www.googleapis.com/oauth2/v1/certs",
        "client_x509_cert_url"=>[REDACTED],
        "universe_domain"=>"googleapis.com"
    ];
}

public static function send($userDeviceDetail, $notification)
{
    $fcmservice = new self();
    $projectId = $userDeviceDetail->projectId;
    $serverApiKey = env('FCM_SERVER_KEY');
    $accessToken = $fcmservice->getAccessToken($serverApiKey);

    $endpoint = "https://fcm.googleapis.com/v1/projects/" . $projectId . '/messages/send';
    $responses = [];// Array to store individual responses

    foreach ($userDeviceDetail->pluck('fcToken') as $token) {
        $notificationType = isset($notification['body']['notificationType']) ? (string)$notification['body']['notificationType'] : null;

        $payload = [
            'message' => [
                'token' => $token,
                'notification' => [
                    'title' => $notification['title'],
                    'body' => $notification['body']['description'],
                ],
            ],
        ];
        $data[] = $payload;
    }
}
```

These keys you will get in your json file which is downloaded just copy and paste it.

9) Now Add Your Project name in this controller file
App/Http/Controllers/Admin/ChatController.php

```

try {
    $user = DB::table('users')
        ->join('users', 'users.id', '=', 'tickets.userId')
        ->select('users.name as userName', 'users.profile', 'tickets.userId', 'tickets.ticketStatus')
        ->where('tickets.id', '=', $req->id)
        ->get();
}
$chatId = $req->id . '_' . $user[0]->userId;
$data = array(
    'chatId' => $chatId,
    'userName' => $user[0]->userName,
    'userProfile' => $user[0]->profile,
    'userId' => $user[0]->userId,
    'ticketId' => $req->id,
    'ticketStatus' => $user[0]->ticketStatus,
);

```

// Use Guzzle or any HTTP client to make requests to Firestore REST API

```

// Check if 'documents' key exists in the API response
$response = json_decode($response->getBody(), true);

if (isset($apiResponse['documents'])) {
    $messages = $apiResponse['documents'];
    usort($messages, function ($a, $b) {
        return strtotime($a['createTime']) - strtotime($b['createTime']);
    });
} else {
    // Handle the case when there are no documents
    $messages = [];
}
return view('pages.chat', compact('messages', 'data'));
} else {
    // Handle the case when there are no documents
    $messages = [];
}

```

```

public function createChat(Request $req)
{
    try {
        $apiEndpoint = "https://firestore.googleapis.com/v1/projects/astroway-diploy/databases/(default)/documents/";
        $postData = [
            'fields' => [
                'message' => [
                    'stringValue' => $req->message,
                    'timestampValue' => Carbon\Carbon::now()->toIso8601String(),
                    'updatedAt' => [
                        'timestampValue' => Carbon\Carbon
                    ],
                    'userRef1' => [
                        'integerValue' => $req->userRef1,
                        'userRef2' => [
                            'integerValue' => $req->userRef2
                        ],
                        'status' => [
                            'stringValue' => 'OPEN'
                        ]
                    ],
                    'client' => [
                        'useDate' => [
                            'dateValue' => Carbon\Carbon::now()
                        ]
                    ]
                ]
            ],
            'messageCount' => 2 || $req->ticketId ? 0 : 1
        ];
        $data = ['ticketStatus' => 'OPEN'];
        $DBtable['tickets']->where('id', '=', $req->ticketId)
            ->update($data);
        $userDeviceDetail = DB::table('user_device')->where('user_id', '=', $req->ticketId)
            ->where('user_id', '=', $req->userId)
            ->select('user_device_details')
            ->get();
        if ($userDeviceDetail->count() > 0) {
            $userDeviceDetail->where('user_id', '=', $req->ticketId)
            ->select('user_device_details')
            ->get();
        }
        if ($userDeviceDetail->count() > 0) {
            $userDeviceDetail->where('user_id', '=', $req->ticketId)
            ->select('user_device_details')
            ->get();
        }
    }
}

```

The default interactive shell is now zsh.
To update your account to use zsh, please run chsh -s /bin/zsh.
For more details, please visit https://support.apple.com/kb/HT208058.
apple-iMac:htdocs apple:~

And After that your firebase setup will be done.

Admin Panel Setup

- 1) Now Open Admin Panel for e.g <https://yourdomain.com/admin>
- 2) And Login in that
- 3) After Login Go to general settings in that go to third party package
And setup your keys .

The screenshot shows two views of the Astroway admin settings page. The top view displays the 'General' tab, which includes fields for Customer App Name (Astroway), App Language (multiple language options like English, Gujarati, Hindi, etc.), Partner App Name (Astrologer App), and various other settings like currency and intro video upload. A red arrow points to the 'ThirdPartyPackage' tab at the top right. The bottom view shows the 'ThirdPartyPackage' tab selected, displaying fields for Agora (Agora Key, Agora Secret, Agora App Certificate, Agora AppId) and Google Bucket (Google Access Key, Google Secret Key). A red arrow also points to the 'General' tab on the left sidebar.

Lets Setup Agora keys First

Agora Keys Setup

- 1) Open <https://console.agora.io/> and sign up
- 2) After sign up you will see the option of create project just create your project .After creating just click on configure

The screenshot shows the Agora console's Project Management interface. On the left is a sidebar with icons for Developers, Docs, API Reference, SDKs, Help, and a user profile. The main area has a header with 'console.agora.io/projects', a message about the old version, and a 'Switch to the new version' link. Below the header is a navigation bar with 'Developers', 'Docs', 'API Reference', 'SDKs', 'Help', 'Support Ticket', and a user dropdown. A warning message 'To secure your account, please verify your phone number.' is displayed. The main content area is titled 'Project Management' and shows a table of projects. The columns are 'Project Name', 'Creation Date', 'Stage', 'Security', 'App ID', and 'Action'. There is one row for 'Test', which was created on '2024-03-07' and is currently 'Testing'. The 'Security' status is 'Enabled'. The 'App ID' field contains a long string of characters. A red arrow points to the 'Action' column for the 'Test' project.

3) You will see the app id and app certificate id Just copy it .

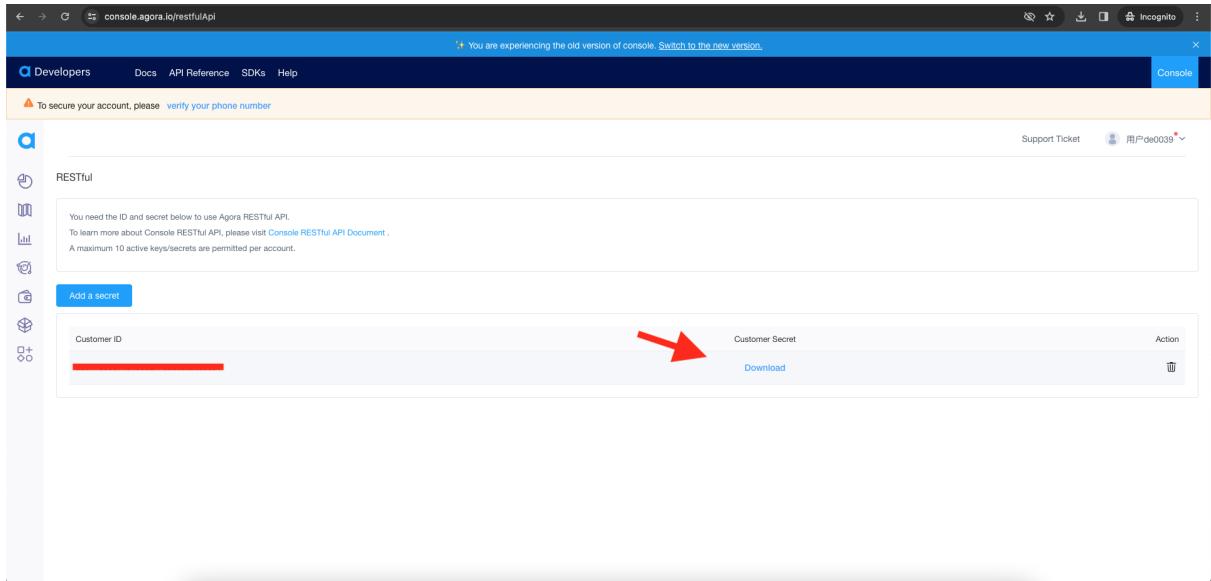
The screenshot shows the Agora console's project configuration page for a project named 'Test'. The top navigation bar includes 'Developers', 'Docs', 'API Reference', 'SDKs', 'Help', 'Support Ticket', and a user dropdown. A warning message 'To secure your account, please verify your phone number.' is present. The main area shows project details: 'Create time: 2024-03-07', 'Project name: Test', and 'Use case: Select'. A red arrow points to the 'App ID' field, which contains a long string of characters. Another red arrow points to the 'App certificate' section, which shows two entries: 'Primary Certificate' (Enabled) and 'Secondary Certificate' (Disabled). The sidebar on the left lists 'Project Management', 'Logs', 'Lifl', 'Logs', 'File', 'Security', and 'Features'.

4) Now Click on RESTful Api .From there you need to create key.

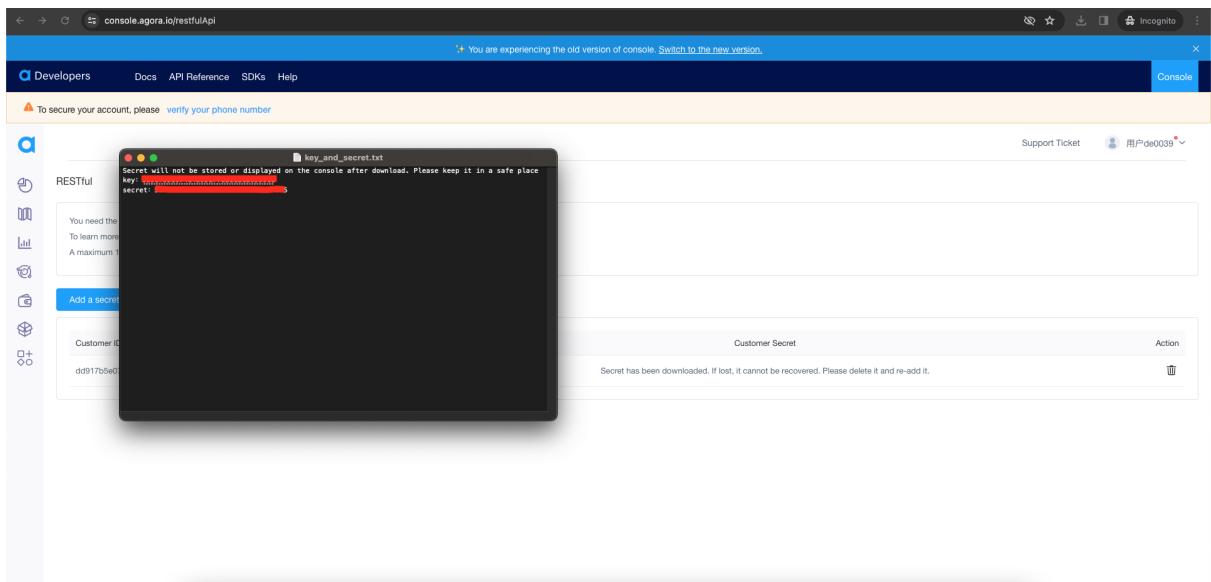
This screenshot shows the Agora console interface for a project named 'Test'. The top navigation bar includes links for Developers, Docs, API Reference, SDKs, and Help. A message at the top right encourages account verification. On the left, a sidebar lists various project settings like Security, Features, and Metrics. The main content area displays project details: Create time: 2024-03-07, Project name: Test, Use case: Select, and App ID: (redacted). In the top right corner, there's a user profile for 'User de0039' with options for Support Ticket, Messages, Settings, and Log out. A red arrow points to the 'RESTful API' link in the user menu.

This screenshot shows the 'RESTful' configuration page. The top navigation bar and sidebar are identical to the previous screenshot. The main content area has a heading 'RESTful' with a note: 'You need the ID and secret below to use Agora RESTful API. To learn more about Console RESTful API, please visit [Console RESTful API Document](#). A maximum 10 active keys/secrets are permitted per account.' Below this is a blue 'Add a secret' button, which is highlighted with a red arrow. Further down, there are sections for 'Customer ID' and 'Customer Secret', both currently empty, with a note: 'You don't have any RESTful API ID and secret. Add one to start.' The right side of the screen shows a table header for 'Action'.

From there you need to create key .your key will be generated after that



6) Download this and open that file in that you will see the app key and secret key



Now Just These All Keys in Agora Section of Admin Panel .
Your Agora Setup Completed ..

Now Setup Google Bucket Key

Google Bucket Keys Setup

- 1) For google bucket go to <https://console.cloud.google.com/storage>
- 2) Signup and after that create project
You will see setting option .go to setting in that open interoperability

The screenshot shows the Google Cloud Storage Settings page for a project named 'My Project 43299'. The 'Access keys for service accounts' section indicates that the project doesn't have any service account HMAC keys. A button to 'CREATE A KEY FOR A SERVICE ACCOUNT' is visible. Below this, the 'User account HMAC' section explains how to authenticate using access keys tied to the user account instead of service accounts. It also notes that the Interoperability API uses the default project for creating buckets and listing them. A message states 'You haven't set a default project for your user account yet' and provides a link to 'SET MINERAL-BALM-416209 AS DEFAULT PROJECT'. The 'Access keys for your user account' section displays two redacted access keys and secrets. A 'CREATE A KEY' button is at the bottom.

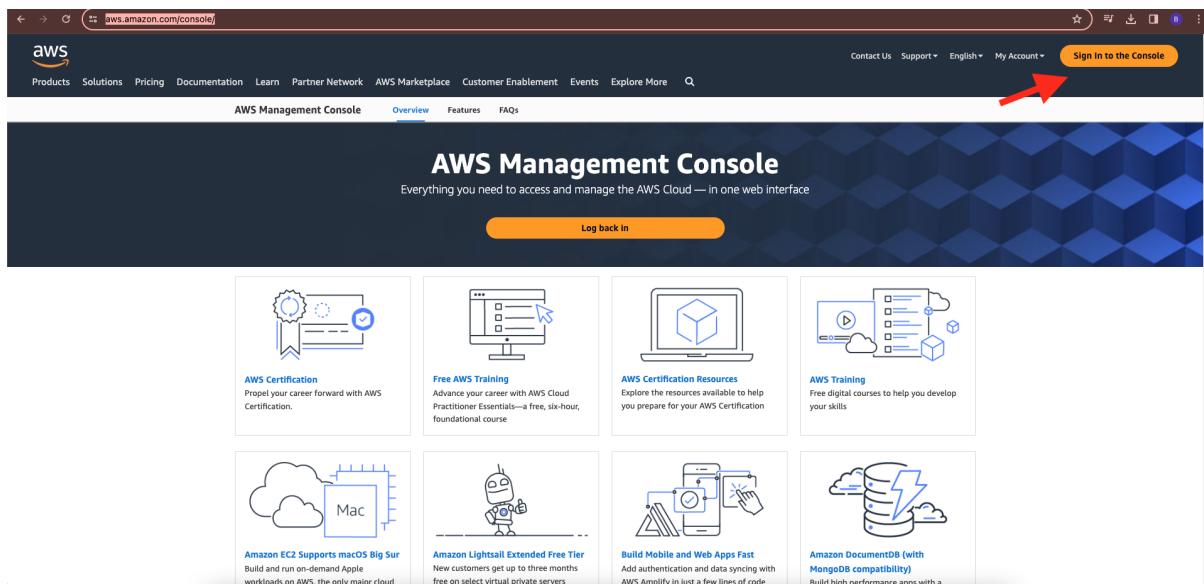
From There you can get key and secret key .now you can update this in your admin panel.

The screenshot shows the 'Astroway' admin panel settings page under the 'ThirdPartyPackage' tab. On the left sidebar, there are various management sections like Blogs, News, Videos, Banner Management, Notifications, Support Management, Earning, Reports, Master Settings, Team Management, General Settings, Feedback, and Page Management. The main content area shows configuration for 'Google Bucket' and 'Horoscope API'. Under 'Google Bucket', fields for 'Google Access Key' and 'Google Secret Key' are shown, both redacted. Under 'Google Bucket Name', a redacted value is listed. Under 'Horoscope API', fields for 'Astrology Api Userid' and 'AstrologyApi Key' are shown, both redacted. A 'Vedic Astrology Api' field contains the value '6e2e66ec-077f-42c7-9e0f-0e0000000000'. A 'Save' button is located in the top right corner.

Now Lets Setup Aws S-3

AWS S3 Setup

1) First Open <https://aws.amazon.com/console/> and sign up.



Now Click on Sign Up And Create Your Aws Account..
After That your need to create a bucket to store data in that ..

A screenshot of the Amazon S3 buckets list. The page header says "Amazon S3". Below it, there's a section titled "Buckets (0)" with a note: "Buckets are containers for data stored in S3. Learn more." To the right of this are buttons for "Copy ARN", "Empty", "Delete", and a prominent orange "Create bucket" button, which is highlighted with a red box. Below this is a search bar with the placeholder "Find buckets by name". Underneath is a table with columns: Name, Region, Access, and Creation date. The table shows "No buckets" and "You don't have any buckets." At the bottom of the table is another "Create bucket" button.

Click On Create Bucket ..

Amazon S3 > Create bucket

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

Region

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose Bucket Name And Region And Create it..

⌚ Successfully created bucket "magecomptest123awsbucket"
To upload files and folders, or to configure additional bucket settings choose [View details](#).

View details X

Amazon S3

Buckets (1)				<input type="button" value=""/>	<input type="button" value="Copy ARN"/>	<input type="button" value="Empty"/>	<input type="button" value="Delete"/>	<input type="button" value="Create bucket"/>
Buckets are containers for data stored in S3. Learn more								
<input type="text" value="Find buckets by name"/>				< 1 > <input type="button" value=""/>				
Name	Region	Access	Creation date					
<input checked="" type="radio"/> magecomptest123awsbucket	Asia Pacific (Seoul) ap-northeast-2	Bucket and objects not public	January 23, 2021, 17:53:01 (UTC+05:30)					

Bucket Created Successfully ..

Now Lets find Key and Secret Key..

Click on My Security Credential

The screenshot shows the AWS S3 console interface. At the top, there's a search bar and a navigation bar with 'Global' and 'Support' options. A dropdown menu is open from the top right, listing 'My Account', 'My Organization', 'My Service Quotas', 'My Billing Dashboard', and 'My Security Credentials'. The 'My Security Credentials' option is highlighted with a red box. Below the menu, there's a 'Create bucket' button. The main content area shows 'Buckets (1)' with a table listing one bucket: 'magecomptest123awsbucket' located in 'Asia Pacific (Seoul) ap-northeast-2'. The table columns are 'Name', 'Region', 'Access', and 'Last Used'. A 'Sign Out' link is also visible.

After That Create New Access Keys

The screenshot shows the 'Access keys' section in the AWS IAM console. A sub-menu header 'Access keys (access key ID and secret access key)' is at the top. Below it, a note says 'Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, the AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.' A 'Learn more' link is provided. A table lists existing access keys, with columns: 'Created', 'Access Key ID', 'Last Used', 'Last Used Region', 'Last Used Service', 'Status', and 'Actions'. A prominent red box highlights the 'Create New Access Key' button at the bottom left of the table.

From There your access key will be generated.

The screenshot shows a 'Create Access Key' confirmation dialog. It displays a green success message: 'Your access key (access key ID and secret access key) has been created successfully.' Below it, a note says 'Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.' To help protect security, it advises storing the secret access key securely and not sharing it. A 'Hide Access Key' link is available. At the bottom, there are fields for 'Access Key ID:' and 'Secret Access Key:', and buttons for 'Download Key File' and 'Close'.

Now You Need To Change Policy and Permission of Bucket

The screenshot shows the AWS S3 console with the 'astroway' bucket selected. The 'Permissions' tab is active. In the 'Block public access (bucket settings)' section, there is a note about granting public access through various methods. Below it, the 'Block all public access' setting is shown as 'OFF' with a warning icon. A red arrow points to this setting. In the 'Bucket policy' section, the JSON policy code is displayed, and another red arrow points to the beginning of the code block.

Make it Access Public ..

And After That Set Policy.

The screenshot shows the same AWS S3 console for the 'astroway' bucket. The 'Permissions' tab is still active. The 'Block all public access' setting is now shown as 'OFF' with a green checkmark, indicating it has been successfully disabled. The 'Bucket policy' section displays the JSON policy code, with a red arrow pointing to the start of the code block.

For E.g It can be like this :

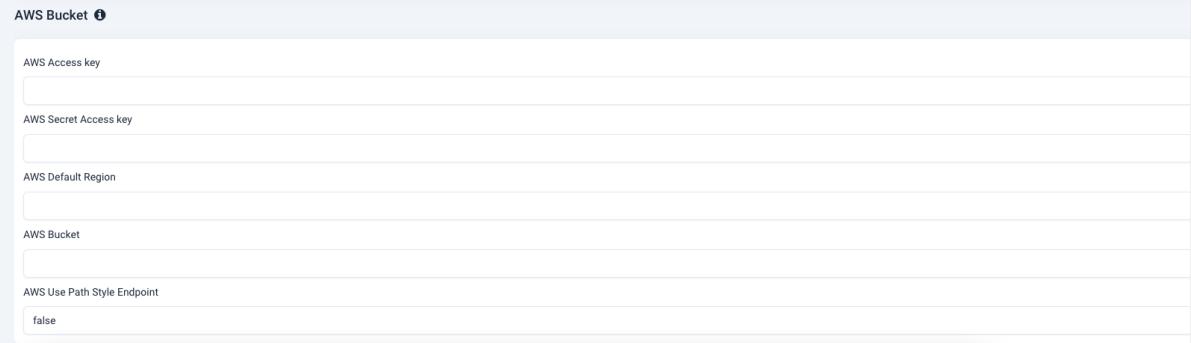
```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::astroway/*"  
    }  
  ]}
```

```
]  
}
```

Change Your bucket name with your bucket ..

Now Your Aws Setup will done Perfectly ..

Now Lets Setup Keys in Admin Panel:



AWS Bucket

AWS Access key

AWS Secret Access key

AWS Default Region

AWS Bucket

AWS Use Path Style Endpoint

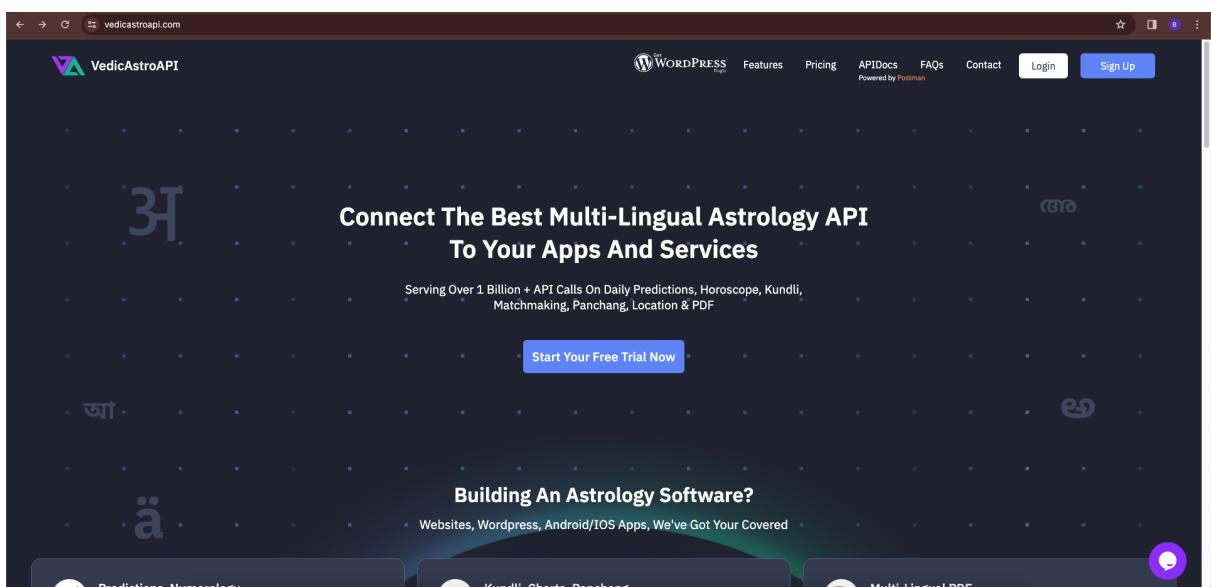
false

Add Your Access Key , Secret Key, Region and Your Bucket name ..

Now Lets Setup Vedic Astro API

Vedic Astro API

1) First Open <https://vedicastroapi.com/> and sign up.



2) After Successfully Signup you will get the vedic astro api key.

The screenshot shows the VedicAstroAPI dashboard. On the left sidebar, there are links for Dashboard, Buy Plan, Usage, Account, Support, Testing, Back To Website, and Log out. The main area has sections for API Calls Used (Topup API) and PDF API Calls Used. The API Key section displays a long string of characters (redacted in the image) with a 'Copy' button. A red arrow points to this 'Copy' button. To the right, there's a One Time Plan section showing 'Freepro' and 'Free' plans, with a 'Renew Plan' button. Below that is a 'Logo For PDF' section with upload tips and a 'Upload Logo In PNG' button. At the bottom, there's a 'March API Usage' section showing a circular chart and a message 'No Data For This Month'.

Now Add this in your admin panel.

The screenshot shows the Astroway admin panel settings page. The left sidebar includes links for Blogs, News, Videos, Banner Management, Notifications, Support Management, Earning, Reports, Master Settings, Team Management, General Settings, Feedback, and Page Management. The main content area is titled 'Settings' and has tabs for General, Payments, SocialLink, ThirdPartyPackage (which is selected), and MasterImage. Under the 'ThirdPartyPackage' tab, there are sections for 'Google Bucket' and 'Horoscope API'. The 'Google Bucket' section contains fields for Google Access Key, Google Secret Key, and Google Bucket Name, all of which are redacted. The 'Horoscope API' section contains fields for Astrology Api Userid, AstrologyApi Key, and Vedic Astrology Api, where the 'Vedic Astrology Api' field is highlighted with a red arrow.

You Also Need To Add this key to your
app/Http/Controllers/API/User/KundaliMatchingController.php

```

    ...
    $data = $req->only([
        'male_kundli_id',
        'female_kundli_id'
    ]);

    $maleKundliId = $req->male_kundli_id;
    $femaleKundliId = $req->female_kundli_id;

    $maleKcd = Kundali::where('id', $maleKundliId)->first();
    $femaleKcd = Kundali::where('id', $femaleKundliId)->first();

    $girlManglikDate = date('Y/m/d', strtotime($maleKcd->birthDate));
    'dob' => $maleKcd->birthTime,
    'tz' => $maleKcd->timeZone,
    'lat' => $maleKcd->latitude,
    'lon' => $maleKcd->longitude,
    'api_key' => '436dd872-b655-5259-aab0-133e20301cf3',
    'lang' => 'en'
]);

```

```

    $boyManglikDate = Https::get('https://api.vedicastroapi.com/v3-json/dosha/manglik-dosh', [
        'dob' => date('Y/m/d', strtotime($femaleKcd->birthDate)),
        'tz' => $femaleKcd->timeZone,
        'lat' => $femaleKcd->latitude,
        'lon' => $femaleKcd->longitude,
        'api_key' => '436dd872-b655-5259-aab0-133e20301cf3',
        'lang' => 'en'
]);

```

```

    if(strtowlower($femaleKcd->match_type) == strtowlower('North')){
        ...
        $dailyHoroscope = Https::get('https://api.vedicastroapi.com/v3-json/matching/ashtakoot', [
            'boy_dob' => date('d/m/Y', strtotime($maleKcd->birthDate)),
            'boy_tz' => $maleKcd->timeZone,
            'boy_lat' => $maleKcd->latitude,
            'boy_lon' => $maleKcd->longitude,
            'girl_dob' => date('d/m/Y', strtotime($femaleKcd->birthDate)),
            'girl_tz' => $femaleKcd->timeZone,
            'girl_lat' => $femaleKcd->latitude,
            'girl_lon' => $femaleKcd->longitude,
            'api_key' => '436dd872-b655-5259-aab0-133e20301cf3',
            'lang' => 'en'
        ]);
    }else{
        ...
        $dailyHoroscope = Https::get('https://api.vedicastroapi.com/v3-json/matching/dashakoot', [
            'boy_dob' => date('d/m/Y', strtotime($maleKcd->birthDate)),
            'boy_tz' => $maleKcd->timeZone
        ]);
    }
}

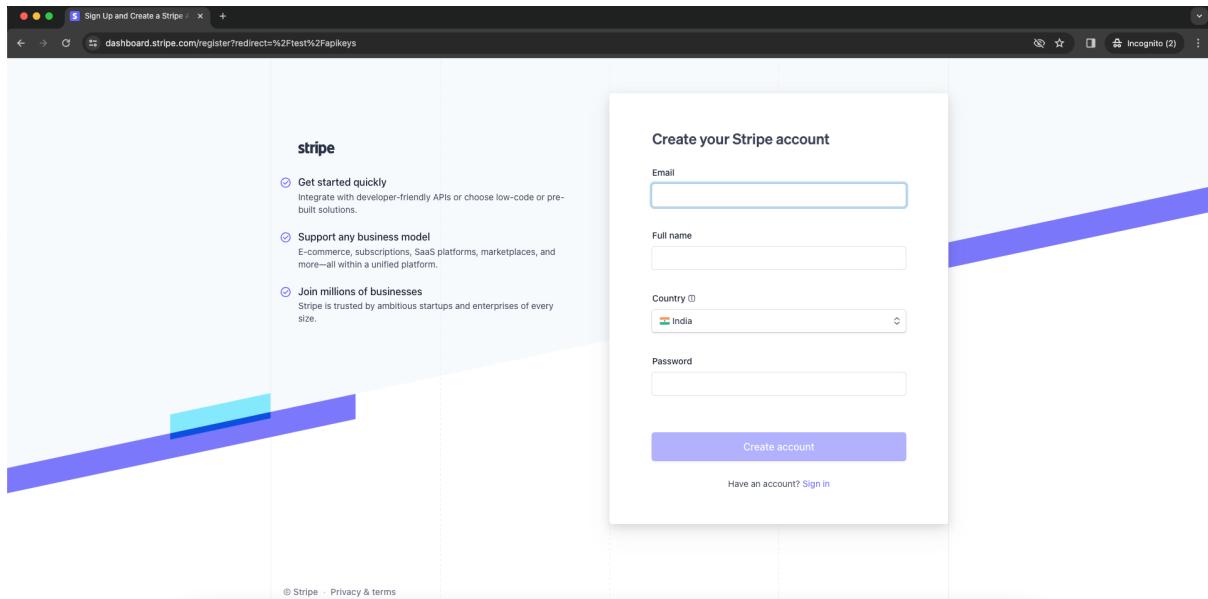
```

Your Third party package setup successfully

Payment Section

Lets Setup Stripe Keys

- 1) You can setup payment keys from payment section For e.g Just go to <https://dashboard.stripe.com/> and sign up.



- 2) After Sign up you can see the test publish and secret key

The screenshot shows the Stripe API keys page under the Developers section. It displays two types of keys: Standard keys and Restricted keys. A red arrow points to the 'Publishable key' row in the Standard keys table. The table has columns for NAME, TOKEN, LAST USED, and CREATED. The 'Publishable key' row contains the token: pk_test_510rgptSATuzK0JfSHr0a9pjfd3u6ZhnyWt89spLh1e2z0sosybz2wpZEnisInavqVotgIBNmMgqDr6nJFZyQoTT088gYRZF9g.

And Add this here

The screenshot shows the Astroway admin settings page under the Payments tab. A red arrow points to the 'Stripe Testing Secret Key' field, which is currently filled with a long redacted string. Below it is the 'Stripe Testing Publish Key' field, also containing a redacted string.

- 3) Similarly you can setup for other payment gateway also . For e.g you can setup razor pay keys by reading their documentation
<https://razorpay.com/docs/>
- 4) In that section there is test mode if its 1 it means its in test mode, if 0 then in live mode set according to your requirement .
- 5) There is around 14 Payment Gateway Setup your keys and all by reading the documentation of particular payment gateway

Cronjob Section

Lets Setup Cron job

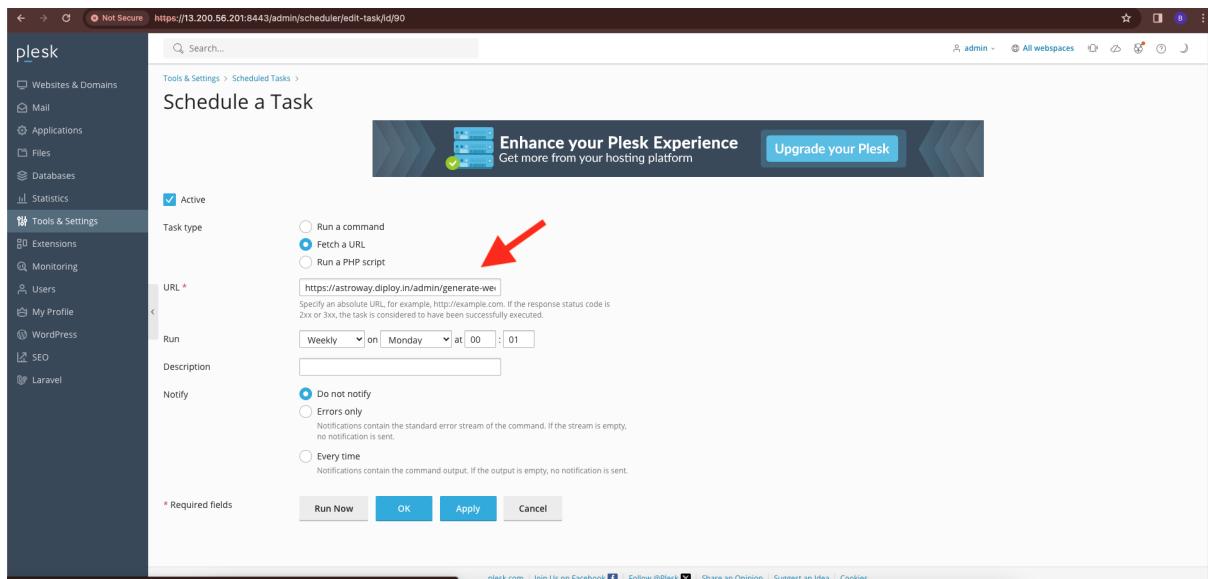
1) Go to Setting of you server .For example i am using plesk i can set like this

Click on schedule task (cron jobs).After that click on add task.

The screenshot shows the Plesk control panel under the 'Tools & Settings' section, specifically the 'Scheduled Tasks' page. A red arrow points to the 'Add Task' button at the top left of the list. The list displays 32 items total, each with a checkbox, a task name, a cron schedule, and a 'Run Now' button. The tasks include various extensions like 'Extension revisium-antivirus', 'Extension monitoring', and 'Extension sslit'. The cron schedules range from hourly to daily.

After click on add task you will have to set cronjobs based on daily and weekly also

The screenshot shows the 'Schedule a Task' dialog box. A red arrow points to the 'URL' input field, which contains the URL 'https://astroway.deploy.in/admin/generate-dal'. The dialog includes fields for 'Task type' (set to 'Fetch a URL'), 'Run' (set to 'Daily at 00:01'), 'Description', 'Notify' (set to 'Do not notify'), and buttons for 'Run Now', 'OK', 'Apply', and 'Cancel'. The Plesk header bar is visible at the top.



Add Your Url for daily and weekly .for example

Daily:

<https://astroway.diploy.in/admin/generate-daily-horscope>

Weekly:

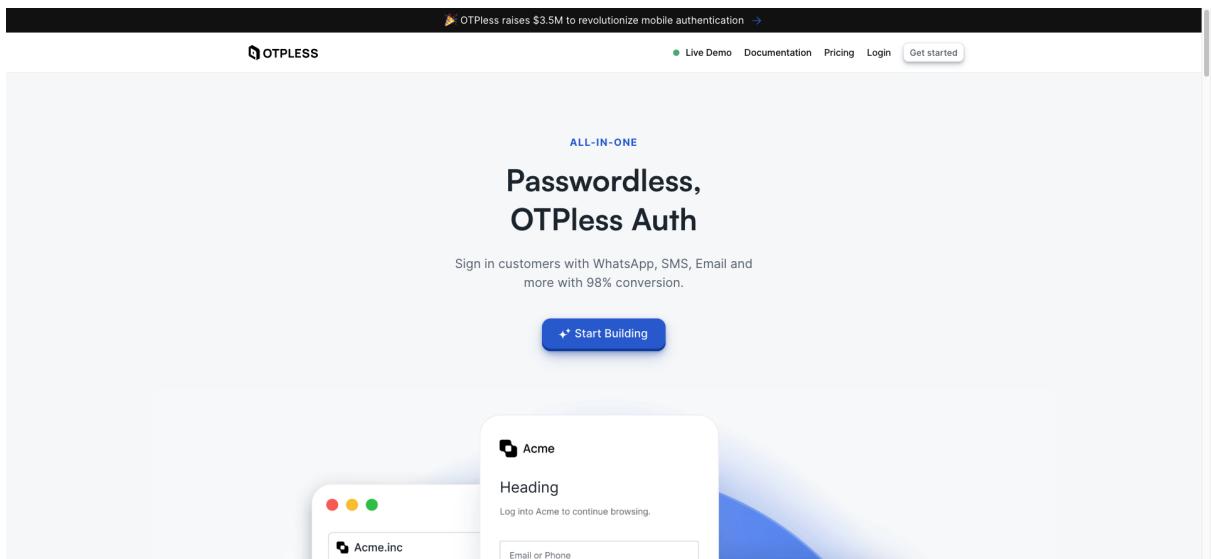
<https://astroway.diploy.in/admin/generate-weekly-horscope>

yearly:

<https://astroway.diploy.in/admin/generate-yearly-horscope>

OTP Less Setup:

1. Go to <https://otpless.com> and Sign In



2. After Sign In go to Configure Channels and Enable Whatsapp , Email Login

The screenshot shows the OTPless dashboard for 'Your First App'. On the left sidebar, 'Configure Channels' is selected and highlighted with a red box. The main panel displays 'Configure Channels' settings for primary methods (Phone Number, Email ID) and social methods (WhatsApp, Google, Apple). A '2 factor Authentication' toggle is also present. To the right, a preview window shows a mobile sign-in screen for 'Astroway' with 'Let's Sign In' and 'Continue' buttons, and social media icons for WhatsApp, Google, and Apple. A 'Customize Design' button is at the bottom of the preview.

3. Now Go to App Setting . From there copy your app id , client id and client secret key and Paste it in Admin Panel.

The screenshot shows the OTPLESS Admin Panel. On the left, there's a sidebar with 'Your First App' selected. Under 'App Settings', the 'General' tab is active, showing fields for 'APP ID', 'Client ID', and 'Client Secret', each with a redacted value. Below these is a section to 'Share credentials with your developer' with a 'Send' button. To the right, a 'Getting started' guide titled 'Your setup' is displayed, showing four steps: 'Inviting your team' (in progress), 'Choose your Auth Method' (completed), 'Integrate to your platform' (completed), and 'Customize your login experience' (in progress). At the bottom right of the panel is a 'Save' button.

4. Go To Your Admin Panel And in General setting > Thirdparty configuration set your keys

The screenshot shows the Astroway Admin Panel. The left sidebar includes options like Dashboard, Customers, Astrologers, Astroshop, Horoscope, Blogs, News, Videos, Banner Management, Stories, Notifications, Support Management, Earning, and Reports. The main area is titled 'Settings' and has tabs for General, Payments, SocialLink, ThirdPartyPackage (which is selected), MasterImage, and WebsiteConfig. Under the 'ThirdPartyPackage' tab, there are sections for 'AWS Use Path Style Endpoint' (with a redacted value), 'Google Map Api' (with a redacted 'Google Map Api Key'), and 'OTP Less' (with fields for 'Otp Less App Id', 'Otp Less Client Id', and 'Otp Less Secret Key', all redacted). A 'Save' button is located at the top right of the settings area.

Google Map Api:

5. You Also Need to add google map api for fetching the location . And Paste it In Admin Side.

The screenshot shows the 'Astroway' application's settings interface. The left sidebar contains navigation links for Dashboard, Customers, Astrologers, Astroshop, Horoscope, Blogs, News, Videos, Banner Management, Stories, Notifications, Support Management, Earning, and Reports. The main content area is titled 'Settings' and has tabs for General, Payments, SocialLink, ThirdPartyPackage (which is selected), MasterImage, and WebsiteConfig. Under the 'ThirdPartyPackage' tab, there are sections for 'astroway', 'AWS Use Path Style Endpoint' (set to false), 'Google Map Api' (highlighted with a red box), 'OTP Less' (highlighted with a red box), and 'MasterImage' (highlighted with a red box). A 'Save' button is located in the top right corner.

Website Configuration

Firebase Configuration For Website:

6. Go to Project Setting/General.
7. Click on Add App.
8. Choose App Name
9. In second step choose script option and copy the details and paste in admin panel.

The screenshot shows the Firebase Project Overview page with a red box highlighting the 'Project settings' button in the top navigation bar. The main content area is titled 'Project settings' and has tabs for General, Cloud Messaging, Integration, Service accounts, Data privacy, and Users and permissions. The 'General' tab is selected. The 'Your project' section displays basic project information: Project name (Astroway), Project ID (astroway-deploy), Project number (381086206621), and Web API key (AlzaSyAyZi-o6qII2X7hNcCgtbmRT2wLAhs). The 'Environment' section indicates 'Unspecified'. The 'Your apps' section lists two Android apps: 'Astroway Partner' (com.astroway.partner) and 'Astroway User' (com.astroway.user). An 'Add app' button is visible in this section. On the left sidebar, there are sections for Generative AI, Build with Gemini, Project shortcuts (Analytics Dashboard, Firestore Database, Events, Authentication, App Check, Storage), Product categories (Build, Run, Analytics), and Blaze (Pay as you go, Modify).

The screenshot shows two parts of the configuration process. The top part is a modal from the Firebase console titled 'Add Firebase to your web app'. It has two tabs: 'Register app' (selected) and 'Add Firebase SDK'. Under 'Add Firebase SDK', there are two options: 'Use npm' (radio button) and 'Use a <script> tag' (radio button, selected). A note says: 'If you don't use build tools, use this option to add and use the Firebase JS SDK. Use this option to get started, but it's not recommended for production apps.' Below is a code snippet for initializing the Firebase configuration:

```

<script type="module">
// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebasejs/10.12.2/firebase-app.js";
import { getAnalytics } from "https://www.gstatic.com/firebasejs/10.12.2/firebase-analytics.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyAvIz-oI0G1L2X-7NnCgzbmRTZNLAhc",
  authDomain: "astroway-diploy.firebaseioapp.com",
  databaseURL: "https://astroway-diploy-default-rtdb.firebaseio.com",
  projectId: "astroway-diploy",
  storageBucket: "astroway-diploy.appspot.com",
  messagingSenderId: "38108620621",
  appId: "1:38108620621:web:c082c6274c92c32e274",
  measurementId: "G-ZHTEXX5N8A"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

```

The bottom part is a screenshot of the 'Astroway' application's settings page. The sidebar shows various sections like Dashboard, Customers, Astrologers, Astroshop, Horoscope, Blogs, News, Videos, Banner Management, Stories, Notifications, Support Management, Earning, and Reports. The main area is titled 'Settings' and contains tabs for General, Payments, SocialLink, ThirdPartyPackage, MasterImage, and WebsiteConfig (which is highlighted with a red box). There are input fields for Firebase Measurement Id, App Id, Messaging Sender Id, Storage Bucket, Project Id, Auth Domain, Database Url, and Api Key. A 'Save' button is at the top right.

For Footer and Static Text Changes:

1. To change footer in customer website, the path for that is ([resources/views/frontend/layout/footer.blade.php](#)).
2. To change footer in astrologer website, the path for that is ([resources/views/frontend/astrologers/layout/footer.blade.php](#))
3. Now to change static text in home page the path is ([resources/views/frontend/pages/index.blade.php](#))

To Update From 1.1 to 2.0 You need to update this:

- 1.replace public/frontend/css/app.css
- 2.replace resources/views/frontend
- 3.and in Database run this command in sql or replace systemflag table

```
INSERT INTO `systemflag` (`id`, `valueType`, `name`, `value`, `isActive`, `isDelete`, `created_at`, `updated_at`, `displayName`, `flagGroupId`, `description`, `parent_id`, `viewenable`) VALUES (NULL, 'MultiSelectWebLang', 'WebLanguage', '["en","gu","hi","mr"]', '1', '0', '2023-04-08 10:21:17', '2023-04-08 10:21:17', 'Web Language', '1', '', '0', '1');
```

Still Have a Question?

Please mail us on nb@diploy.in

Contact Support

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 Team, your go-to hub for top-notch Flutter app development and web applications.

Visit us at <https://diploy.in/>. Thank you for considering us!