**1. Razorpay account:**

* **Sign up/Login to Razorpay Dashboard:** Go to the Razorpay website and sign up for an account or log in if you already have one.
* **Access Dashboard:** Once logged in, you'll be directed to the Razorpay Dashboard. Here, you can manage your payments, view analytics, and access API keys.
* **Navigate to Settings:** Look for the settings or profile section in the Razorpay Dashboard. This is usually located in the top-right corner of the dashboard.
* **Find API Keys:** Within the settings or profile section, there should be an option for API Keys or Integration. Click on this to access your API keys.
* **Generate API Keys:** In the API Keys section, you'll typically find options to generate both test and live API keys. Test keys are used for development and testing purposes, while live keys are used in production.
* Click on the option to generate test keys. This will provide you with a test Key ID and Key Secret.Similarly, generate live keys for use in your production environment.
* **Copy Keys:** Once the keys are generated, they will be displayed on the screen. Copy both the Key ID and Key Secret for both test and live environments. These keys are sensitive, so keep them secure.
* **Set Environment Variables:** Open your development environment or .env file where you manage environment variables.
* Set the RAZORPAY\_KEY\_ID, RAZORPAY\_KEY\_SECRET variables with the test keys you obtained.
* **Configure Environments:** Ensure that your development and production environments are properly configured to use the appropriate set of keys based on the environment. For example, in development, you might use test keys, while in production, you'd use live keys.
* **Integration:** Use these environment variables in your code to dynamically fetch the appropriate Razorpay API keys based on the environment.
* Ensure that your code is using test keys during development and live keys in production to avoid processing real transactions during testing.

**2. payUmoney account:**

* **Sign up/Login to PayUmoney Dashboard:** Go to the PayUmoney Merchant Dashboard (https://www.payumoney.com/) and sign up for an account or log in if you already have one.
* **Access Merchant Dashboard:** After logging in, navigate to the merchant dashboard or settings section. This is where you'll find the credentials needed for integrating PayUmoney's payment gateway.
* **Generate API Keys:**In the settings section, look for options to generate API keys for both the test and live environments. PayUmoney usually provides separate keys for testing and production purposes.
* Generate test keys for development and testing purposes. These keys are used to simulate transactions without any real money being involved.Similarly, generate live keys for use in your production environment, where real transactions will occur.
* **Copy Keys:** Once the keys are generated, PayUmoney will typically display them on the screen. You'll usually get both the merchant key and the merchant salt.Copy both the merchant key and merchant salt for both test and live environments. Keep these keys secure, as they are sensitive information.
* **Set Environment Variables:** Open your development environment or .env file where you manage environment variables.
* Set the PAYUMONEY\_MERCHANT\_KEY, PAYUMONEY\_MERCHANT\_SALT variables with the live keys you obtained.
* **Configure Environments:** Ensure that your development and production environments are properly configured to use the appropriate set of keys based on the environment. For example, in development, you might use test keys, while in production, you'd use live keys.
* **Integration:** Use these environment variables in your code to dynamically fetch the appropriate PayUmoney merchant keys based on the environment.Ensure that your code is using test keys during development and live keys in production to avoid processing real transactions during testing.

**3. paytm account:**

* **Sign up/Login to Paytm Dashboard:** Go to the Paytm Merchant Dashboard (https://dashboard.paytm.com/merchant/login) and sign up for an account or log in if you already have one.
* **Access Developer Settings:** After logging in, navigate to the developer settings section. This is where you'll find the credentials needed for integrating Paytm's payment gateway.
* **Generate API Keys:** In the developer settings, look for options to generate API keys for both the test and live environments. Paytm usually provides separate keys for testing and production purposes.
* Generate test keys for development and testing purposes. These keys are used to simulate transactions without any real money being involved.Similarly, generate live keys for use in your production environment, where real transactions will occur.
* **Copy Keys:** Once the keys are generated, Paytm will display them on the screen. Typically, you'll get both the merchant key and the merchant salt.Copy both the merchant key and merchant salt for both test and live environments. Keep these keys secure, as they are sensitive information.
* **Set Environment Variables:** Open your development environment or .env file where you manage environment variables.
* Set the PAYTM\_MERCHANT\_KEY, PAYTM\_MERCHANT\_SALT variables with the live keys you obtained.
* **Configure Environments:** Ensure that your development and production environments are properly configured to use the appropriate set of keys based on the environment. For example, in development, you might use test keys, while in production, you'd use live keys.
* **Integration:** Use these environment variables in your code to dynamically fetch the appropriate Paytm merchant keys based on the environment.Ensure that your code is using test keys during development and live keys in production to avoid processing real transactions during testing.

**4. paypal account:**

* **Sign up/Login to PayPal Developer Dashboard:** Go to the PayPal Developer Dashboard (https://developer.paypal.com/) and sign up for an account or log in if you already have one.
* **Create an Application:** Once logged in, navigate to the "My Apps & Credentials" section.Click on "Create App" or a similar option to create a new application.
* **Choose Environment:** During the application creation process, you'll be prompted to choose between the Sandbox (for testing) and Live environments.Select the Sandbox environment for testing purposes and the Live environment for production.
* **Generate API Credentials:** After creating the application, PayPal will generate API credentials for you. These include a client ID and a secret key.
* For the Sandbox environment, you'll get sandbox client ID and sandbox secret. For the Live environment, you'll get live client ID and live secret.
* **Copy Credentials:** Once the credentials are generated, PayPal will display them on the screen. You'll usually see both the client ID and the secret key.Copy both the client ID and secret for both Sandbox and Live environments. Keep these credentials secure, as they are sensitive information.
* **Set Environment Variables:** Open your development environment or .env file where you manage environment variables.
* Set the PAYPAL\_CLIENT\_ID, PAYPAL\_SECRET variables with the live credentials you obtained.
* **Configure Environments:** Ensure that your development and production environments are properly configured to use the appropriate set of credentials based on the environment. For example, in development, you might use sandbox credentials, while in production, you'd use live credentials.
* **Integration:** Use these environment variables in your code to dynamically fetch the appropriate PayPal API credentials based on the environment.Ensure that your code is using sandbox credentials during development and live credentials in production to avoid processing real transactions during testing.

**5. strip account:**

* **Create a Stripe Account:** If you haven't already, sign up for a Stripe account at Stripe's website.
* **Log in to Stripe Dashboard:** Once you have an account, log in to the Stripe Dashboard using your credentials.
* **Navigate to API Keys:** In the Dashboard, navigate to the "Developers" section. You can usually find it in the left sidebar menu.
* **View API Keys:** In the Developers section, click on "API Keys" or "Get your test API keys". You'll see a section labeled "Standard keys" or "API keys" which contains your publishable and secret API keys.
* **Copy the Secret Key:** You'll find your STRIPE\_API\_KEY in the section labeled "Secret key". Copy the value of the secret key.
* **Set Environment Variable:** Now that you have your Stripe API key, you need to set it as an environment variable in your system or application environment.If you're deploying your application, you can set the environment variable in your deployment configuration or hosting platform. Consult the documentation of your hosting provider for instructions on how to set environment variables.
* **Accessing Environment Variable in Code:** Once the environment variable is set, your application can access it using the process.env object. In your code, you already have const { NODE\_ENV, STRIPE\_API\_KEY } = process.env;, which retrieves the value of STRIPE\_API\_KEY from the environment variables.