

## Documentation

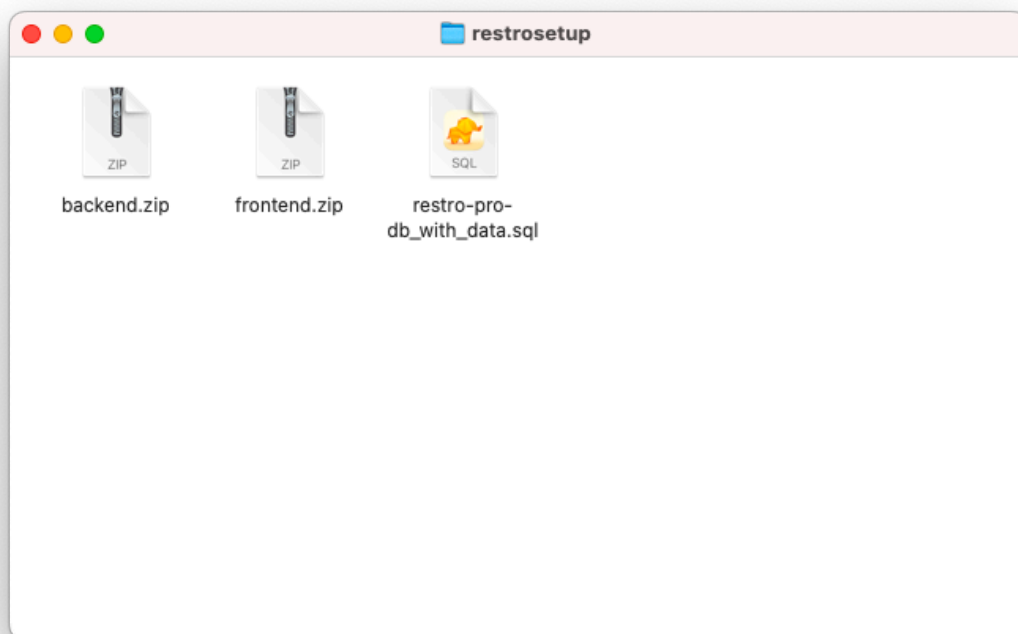
1. [Prerequisites](#)
2. [Creating Stripe Subscription](#)
3. [Server Setup & Database config](#)
4. [Backend code Deployment](#)
5. [Frontend code Deployment](#)
6. [Configuring RestroPro for your outlet](#)
  - 6.1: [App Config](#)
  - 6.2: [Setting Up Super Admin](#)

## 7. [Installation & Customisation Service](#)

### 1. Prerequisites

RestroPRO, the product is built using following technologies NodeJS, MySQL, ReactJS.






So to Setup the project, you'll need technical knowledge and, you'll need to purchase a Ubuntu Server and domain. Before doing that, unzip the provided files, and keep the source code zip file ready.



### 2. Stripe Subscription Setup

Goto stripe.com and set up your account according to your business registration type. Once you activate your account, follow the given procedure.

Goto Stripe Dashboard and Select "Product catalogue", then Add new product.

-  Home
-  Balances
-  Transactions
-  Customers
-  **Product catalogue**

Feel the details as per your business requirements.

### Add a product Close preview ×

**Name (required)**  
Name of the product or service, visible to customers.

**Description**  
Appears at checkout, on the customer portal, and in quotes.

**Image**  
Appears at checkout. JPEG, PNG or WEBP under 2MB.

[Upload](#)

[More options](#) ▾

**Recurring**  One-off

**Amount (required)**

 USD ▾

**Billing period**

 ▾

[More pricing options](#)

### Preview

Estimate totals based on pricing model, unit quantity, and tax.

**Unit quantity**

---

1 × US\$10.00 = **US\$10.00**

---

Subtotal	<b>US\$10.00</b>
Tax	<a href="#">Start collecting tax</a>
Total per month	<b>US\$10.00</b>

Billed at the start of the period

Once you create the Subscription Product, you will see it like this.

Q Search Developers Test mode  ? 🗨 ⚙ +

## Product catalogue + Add product

[All products](#) [Features](#) [Coupons](#) [Shipping rates](#) [Tax rates](#) [Pricing tables](#) [Meters](#)

All  
2
Active  
2
Archived  
0

⊕ Created ⊕ Status 📄 Export prices 📄 Export products ⚙ Edit columns

Name	Pricing	Created	Updated	
RestroPRO Month	US\$5.00 🔄 Per month	7 Jun	7 Jun	...
RestroPRO Subscription	US\$5.00 🔄 Per month	5 Jun	5 Jun	...

2 results

Click on your created Subscription Product. Look for Pricing, and click on options three dots, look for option to copy price id, copy it and save temporary in notes.

Q Search Developers T€

[Products](#) >

### RestroPRO Subscription Active

US\$5.00 • 🔄 Per month

### Pricing +

Price	Subscriptions	Created	
US\$5.00 🔄 Per month	Default 0 active	5 Jun	...

1 results

### Cross-sells

Suggest a related product for customers to add to their order, directly in Checkout. [Learn more.](#)

Cross-sells to

### Features

No features

Copy price ID

Pricing

Set as default price

[Edit price](#)

[Archive price](#)

[Delete price](#)

---

Accept Payments

[Create payment link](#)

[Create new pricing table](#)

Now open the front-end code. And navigate to `src/config/config.jsx`, open the file and look at the highlighted part, we need to edit the default values there. First change the `subscriptionPrice` to the price that you decided for the subscription. Then replace the price id.

```
1 export const API = import.meta.env.VITE_BACKEND
2 export const VITE_BACKEND_SOCKET_IO = import.meta.env.VITE_BACKEND_SOCKET_IO
3
4 export const iconStroke = 1.5;
5
6 export const subscriptionPrice = "$5";
7 export const stripeProductSubscriptionId = "price_1P0sjYSCwICS3BoQN2lnBMTz";
```

Now go back to Stripe Dashboard and open the developers page, you will get API keys from there. Look for a secret key, and copy it.

**Developers**

Overview **API keys** Webhooks Events Logs Apps

**API keys** [Learn more about API authentication](#)

**Standard keys**  
Create a key that unlocks full API access, enabling extensive interaction with your account. [Learn more](#)

NAME	TOKEN	LAST USED	CREATED
Publishable key	[REDACTED]	10 Jun	12 May
<u>Secret key</u>	[REDACTED]	11 Jun	13 May

**Restricted keys**  
Create a key with specific access limits and permissions for greater security. [Learn more](#) [+ Create restricted key](#)

NAME	TOKEN	LAST USED	CREATED
No restricted keys			

Now open the backend code.

```
.env -- restropro-saas-backend
1 DATABASE_URL='mysql://root:12345678@localhost:3306/restropro_saas'
2 JWT_SECRET=restro_jwt_secret
3 JWT_EXPIRY=15m
4 JWT_EXPIRY_REFRESH=30d
5
6 # provide in ms - milliseconds => 15 minute, 30days | match the value to JWT
  expiry
7 COOKIE_EXPIRY=300000 # 15 minute
8 COOKIE_EXPIRY_REFRESH=2592000000 # 30 day
9
10 PASSWORD_SALT=10
11 FRONTEND_DOMAIN="http://localhost:5173"
12 FRONTEND_DOMAIN_COOKIE="localhost"
13
14 STRIPE_SECRET=[REDACTED]
15 STRIPE_WEBHOOK_SECRET=[REDACTED]
16
17 SMTP_HOST=[REDACTED]
18 SMTP_PORT=[REDACTED]
19 SMTP_EMAIL=[REDACTED]
20 SMTP_PASSWORD=[REDACTED]
```

Paste the secret code in STRIPE\_SECRET

Now we will set up the webhook. For that open Stripe Dashboard and goto developers page again. Goto Webhooks Tab, then click on Add endpoint.

## Developers

Overview API keys **Webhooks** Events Logs Apps

Try Workbench View, create, inspect, and edit your webhooks from anywhere in the dashboard.

[Learn more](#) X

### Hosted endpoints

[+ Add endpoint](#)

Then provide the webhook URL, this will be our backend route webhook handler, so we're yet to setup the backend, so you just need to provide the url. For example if i'm setting Backend on domain "api.example.com" then the webhook URL will look like this.

Webhook URL: <https://api.example.com/api/v1/auth/stripe-webhook>

Replace the domain, to your backend domain. After placing endpoint URL, click on the 'select events to listen to'

## Listen to Stripe events

Add an endpoint

Test in a local environment

Set up your webhook endpoint to receive live events from Stripe or [learn more about webhooks](#).

Endpoint URL

https://api.example.com/api/v1/auth/stripe-webhook

Description

An optional description of what this webhook endpoint is used for...

Listen to

Events on your account  Events on Connected accounts ⓘ

Select events to listen to

+ Select events

Add endpoint

Cancel

Look for the following events and select them.

customer.subscription.created

customer.subscription.deleted

customer.subscription.paused

customer.subscription.pending\_update\_applied

customer.subscription.pending\_update\_expired

customer.subscription.resumed

customer.subscription.trial\_will\_end

×

customer.subscription.updated

×

Once you add it open the created webhook endpoint in Stripe Dashboard.

## Developers

Overview API keys **Webhooks** Events Logs Apps

Try Workbench View, create, inspect, and edit your webhooks from anywhere in the dashboard.

Learn more X

### Hosted endpoints

+ Add endpoint

URL	TYPE	LAST 7 DAYS	ERROR RATE	STATUS
https://[redacted]/api/v1/auth/stripe-webhook	Account		24.32%	Active

Click on the 'Reveal' to copy the Webhook Signing Secret.

### Webhooks

https://[redacted]/api/v1/auth/stripe-webhook

Status: Enabled  
Listening for: 8 events  
API version: 2024-04-10  
Signing secret: **Reveal**  
Configuration: View logs

Now go back to the backend .env file. And paste the copied secret in STRIPE\_WEBHOOK\_SECRET

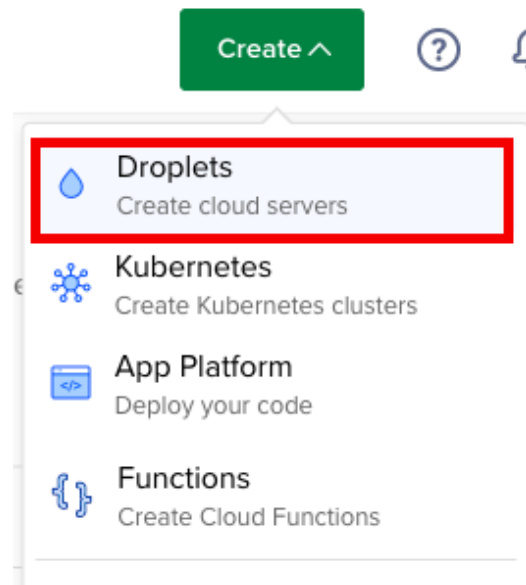
```
.env — restropro-saas-backend
1 DATABASE_URL='mysql://root:12345678@localhost:3306/restropro_saas'
2 JWT_SECRET=restro_jwt_secret
3 JWT_EXPIRY=15m
4 JWT_EXPIRY_REFRESH=30d
5
6 # provide in ms - milliseconds => 15 minute, 30days | match the value to JWT
  expiry
7 COOKIE_EXPIRY=300000 # 15 minute
8 COOKIE_EXPIRY_REFRESH=259200000 # 30 day
9
10 PASSWORD_SALT=10
11 FRONTEND_DOMAIN="http://localhost:5173"
12 FRONTEND_DOMAIN_COOKIE="localhost"
13
14 STRIPE_SECRET=[redacted]
15 STRIPE_WEBHOOK_SECRET=[redacted]
16
17 SMTP_HOST=[redacted]
18 SMTP_PORT=[redacted]
19 SMTP_EMAIL=[redacted]
20 SMTP_PASSWORD=[redacted]
```

Extra setup for sending forgot password email:  
Paste your SMTP configuration in the .env file

### 3. Server Setup & Database config

We'll be using DigitalOcean to setup the server, however you can use any Provider which provides Ubuntu server.

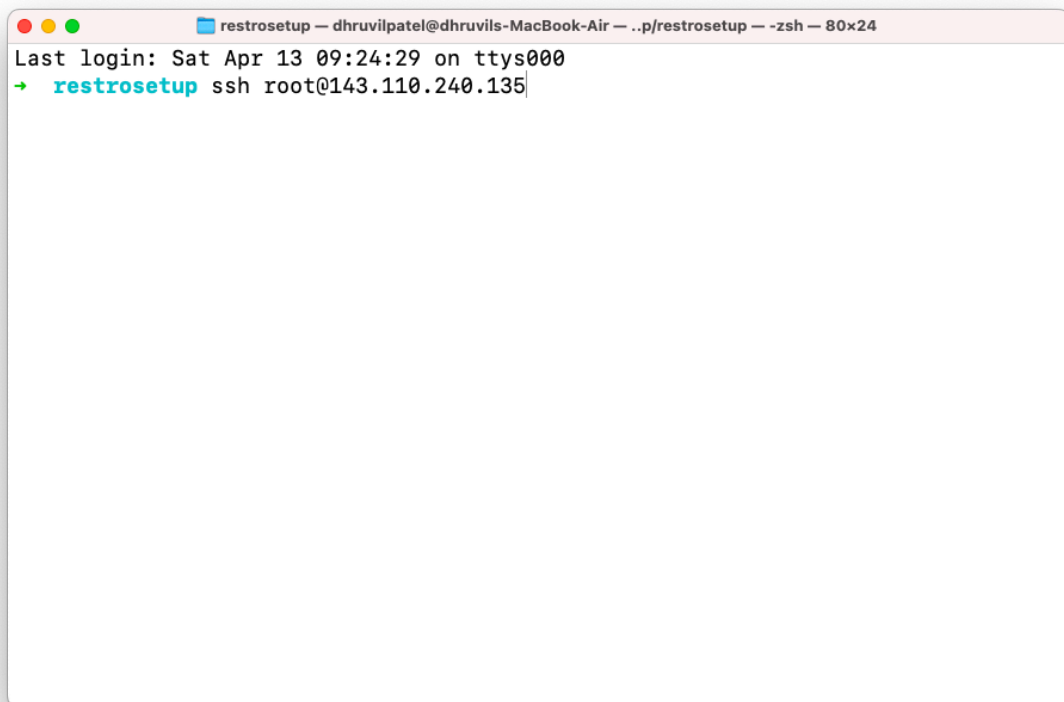
Login to digitalocean or signup. After login you will need to create a droplet (ubuntu server). Click on the Create button and then select "Droplets".



Select the Ubuntu server and complete the process. Make sure you select the SSH method for signing into the droplet, it will ask you to put your device's SSH key, so you can easily and securely SSH into your cloud server.

After the completion of the creation process, open the Terminal on your computer. And type the following command.

```
> ssh root@your_server_ip
```



```
restrosetup — dhruvilpatel@dhruvils-MacBook-Air — ..p/restrosetup — -zsh — 80x24
Last login: Sat Apr 13 09:24:29 on ttys000
→ restrosetup ssh root@143.110.240.135
```

Then You'll be prompted for your SSH key passphrase, just provide it and you'll be connected to your server.



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 80x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

Now update the server by running the following command.

```
> sudo apt update
```

A terminal window titled "restrosetup" showing a root user on an Ubuntu server. The command "sudo apt update" is entered and highlighted in black. The terminal output is currently blank.

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 80x24  
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo apt update
```

Once the update process is completed we'll begin installing the MySQL server.

```
> sudo apt install mysql-server
```

A terminal window with a title bar that reads "restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 80x24". The terminal content shows the prompt "root@ubuntu-s-1vcpu-1gb-blr1-01:~#" followed by the command "sudo apt install mysql-server" which is highlighted in black. The rest of the terminal is empty.

```
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo apt install mysql-server
```

Once the MySQL server is installed run this command to start it.

```
> sudo systemctl start mysql.service
```

A terminal window with a title bar that reads "restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 80x24". The terminal content shows the prompt "root@ubuntu-s-1vcpu-1gb-blr1-01:~#" followed by the command "sudo systemctl start mysql.service" which is highlighted in black with white text.

You can now login to mysql server by writing the following command.  
Make sure when password is prompted the first time, just hit enter, we'll reset the root password in next steps.

```
> mysql -u root -p
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 80x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo systemctl start mysql.service
root@ubuntu-s-1vcpu-1gb-blr1-01:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.23.10.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

To reset the password paste following SQL command. And make sure to put your own password.

```
> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'MYSECRET';
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 91x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36-0ubuntu0.23.10.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'MYSECRET';
```

Now we'll create a database. Use the following command to create it.

```
> CREATE DATABASE restropro;
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 91x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.36-0ubuntu0.23.10.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE restropro;
Query OK, 1 row affected (0.00 sec)

mysql> |
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 91x24
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE restropro;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema     |
| restropro               | ← Our created database
| sys                     |
+-----+
5 rows in set (0.01 sec)

mysql> |
```

Now close the connection to the server.

Now we'll import the database backup file, the database backup file includes structure and sample data pre configured.

```
restrosetup -- dhruvilpatel@dhruvils-MacBook-Air -- ../restrosetup -- ssh -- 91x24
→ restrosetup ls
backend.zip          frontend.zip        resto-pro-db_with_data.sql
→ restrosetup |
```

Copy the highlighted file to the server from your computer. To do that run the following command.

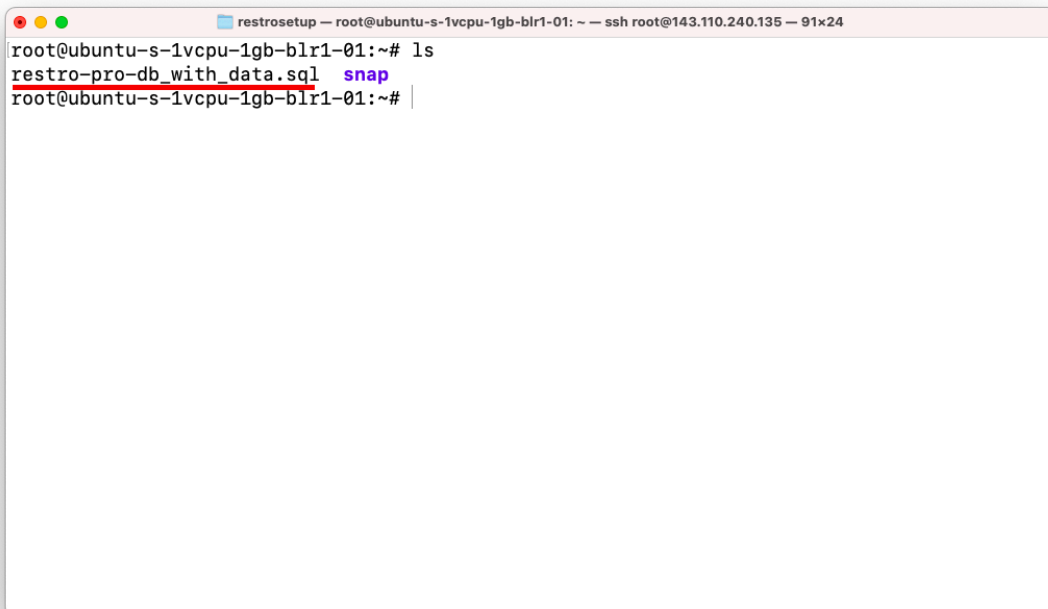
```
> scp resto-pro-do_with_data.sql root@YOUR_SERVER_IP:~/
```

Replace the YOUR\_SERVER\_IP with actual IP ADDRESS



```
restrosetup — dhruvilpatel@dhruvils-MacBook-Air — ..p/restrosetup — -zsh — 91x24
→ restosetup ls
backend.zip          frontend.zip          resto-pro-db_with_data.sql
→ restosetup scp resto-pro-db_with_data.sql root@143.110.240.135:~/
```

Now login to your server using the SSH command. And just hit the “ls” command to see if the file is copied or not.



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 91x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# ls
resto-pro-db_with_data.sql snap
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

On server, paste the following command to import the database backup

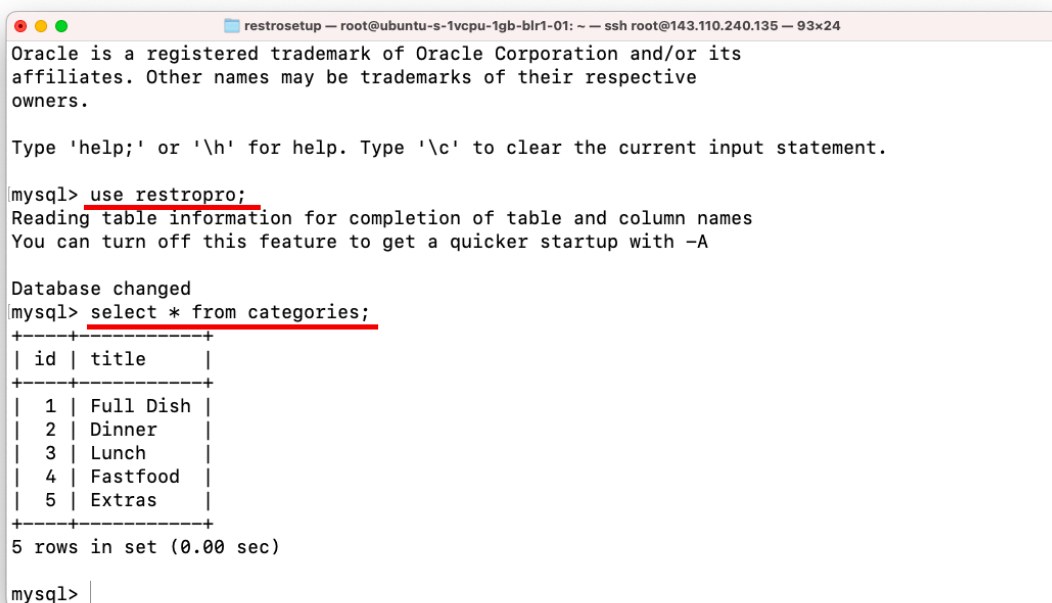
```
> mysql -u root -p restropro < restro-pro-db_with_data.sql
```

Once you hit enter it will ask for your MySQL server password, enter it, and your database import will be completed.



```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# mysql -u root -p restropro < restro-pro-db_with_data.sql
```

You can verify the data by login into MySQL server.



```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 93x24
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use restropro;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from categories;
+----+-----+
| id | title  |
+----+-----+
|  1 | Full Dish |
|  2 | Dinner  |
|  3 | Lunch   |
|  4 | Fastfood |
|  5 | Extras  |
+----+-----+
5 rows in set (0.00 sec)

mysql>
```

So now our Database Server setup is completed, and we'll proceed for Backend code deployment.

## 4. Backend code Deployment

First exit the server by sending 'exit' command in the terminal. As we have to copy the backend zip code from the computer to the Server.

A terminal window screenshot showing the execution of the 'exit' command. The window title is 'restrosetup - dhruvilpatel@dhruvils-MacBook-Air - ..p/restrosetup - -zsh - 93x24'. The terminal output shows: 'root@ubuntu-s-1vcpu-1gb-blr1-01:~# exit', 'logout', and 'Connection to 143.110.240.135 closed.' The prompt is now 'restrosetup |' with a cursor.

```
restrosetup - dhruvilpatel@dhruvils-MacBook-Air - ..p/restrosetup - -zsh - 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# exit
logout
Connection to 143.110.240.135 closed.
→ restrosetup |
```

Use the following command to copy backend zip file to server.

```
> scp backend.zip root@YOUR_SERVER_IP:~/
```

```
restrosetup — dhruvilpatel@dhruvils-MacBook-Air — ..p/restrosetup — -zsh — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# exit
logout
Connection to 143.110.240.135 closed.
→ restrosetup ls
backend.zip          frontend.zip          resto-pro-db_with_data.sql
→ restrosetup scp backend.zip root@143.110.240.135:~/|
```

Once the file is successfully sent to the server, login to the server by using ssh command.

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# ls
backend.zip  resto-pro-db_with_data.sql  snap
root@ubuntu-s-1vcpu-1gb-blr1-01:~# |
```

First thing we will do is unzipping the backend.zip, for that use the following commands.

```
> sudo apt-get install unzip
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# ls
backend.zip  restro-pro-db_with_data.sql  snap
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo apt-get install unzip
```

> unzip backend.zip

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# unzip backend.zip |
```

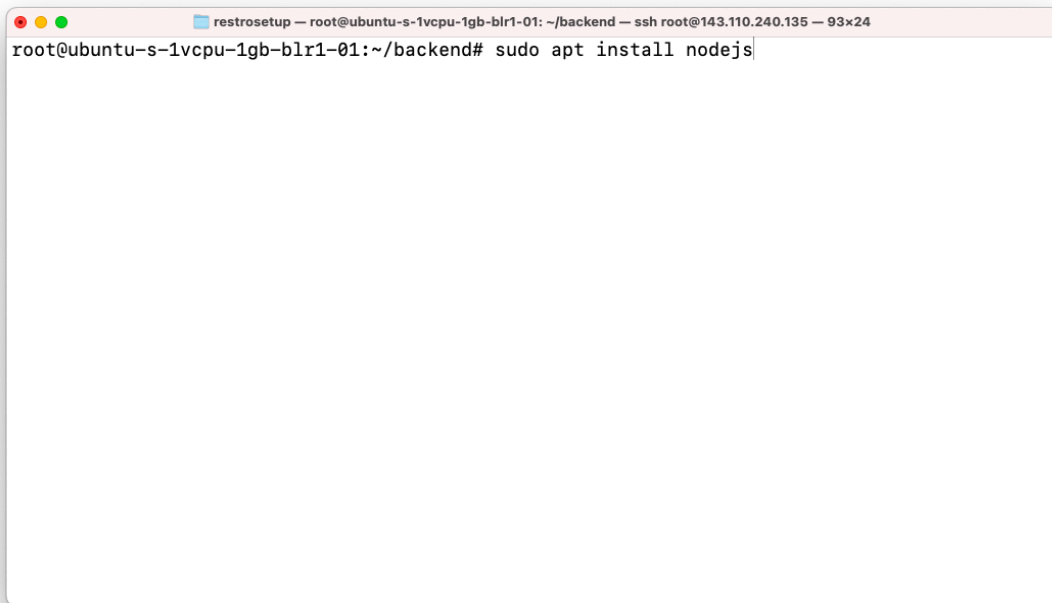
```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# ls
__MACOSX backend backend.zip resto-pro-db_with_data.sql snap
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

> cd backend

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# cd backend
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# ls
index.js package-lock.json package.json src
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend#
```

Now install nodejs using the following command.

> sudo apt install nodejs



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# sudo apt install nodejs|
```

You can check the installation by running the “node -v” command.



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# node -v
v18.13.0
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# |
```

We will also need to install the “npm”. So to install that use the following command.

```
> sudo apt install npm
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend -- ssh root@143.110.240.135 -- 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# sudo apt install npm
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend -- ssh root@143.110.240.135 -- 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# npm -v
9.2.0
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend#
```

We've successfully installed NodeJS and NPM. it's the perfect time that we map the domain to the Server's IP.

For mapping the domain, open the DNS settings available in your domain provider like GoDaddy, Name.com, Hostgator, Bigrock, etc.

And add 'A' Record pointing to **your server's IP**. in the Name value if you want to host the backend on a root URL like myapi.com then write "@".

Else you can use subdomain, for that in the name field write the subdomain, and your server will point to subdomain, in this case i'm linking the Backend server first so in the name i provided "demobackend", this will result in following URL "demobackend.mydomain.com".

Similarly you can also add Another "A" Record for pointing to Frontend. Use the Same IP in the Value Field of your Server. And change the name field to something like "myrestroapp". Anyways for frontend configuration we'll cover this again.

The screenshot shows a web interface for managing DNS records. At the top, there are tabs for 'Overview', 'DNS', and 'Products'. Under the 'DNS' tab, there are sub-tabs: 'DNS Records', 'Forwarding', 'Nameservers', 'Premium DNS', 'Hostnames', 'DNSSEC', and 'NEW'. Below the sub-tabs, there is a description: 'DNS records define how your domain behaves, like showing your website content and delivering your email.' Underneath, there is a section titled 'New Records' with a description: 'A records use an IP address to connect your domain to a website. They're also used to create subdomains such as www or store, that point to an IP address.' Below this description is a form with four fields: 'Type' (set to 'A'), 'Name' (set to 'demobackend'), 'Value' (set to '143.110.240.135'), and 'TTL' (set to '1/2 Hour'). There is a '+ Add another value' link below the 'Value' field. At the bottom of the form, there are three buttons: 'Add More Records', 'Save', and 'Cancel'.

Type *	Name *	Value *	TTL
A	demobackend	143.110.240.135	1/2 Hour

[+ Add another value](#)

[Add More Records](#) [Save](#) [Cancel](#)

Now back to the server, we will install the pm2, for running nodejs apps in the production environment.

```
> sudo npm install pm2@latest -g
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# sudo npm install pm2@latest -g
```

Now we will install the backend code's packages, run the following command.

```
> npm install
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# ls
index.js package-lock.json package.json src
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# npm install
```

After installing the packages, open the .env file, we will edit a few details there.

```
> nano .env
```



We will modify 4 Environment variables, keep rest as it is. Those four environment variables are highlighted in Screenshot.

```
DATABASE_URL="mysql://root:MYPASSWORD@127.0.0.1:3306/restropro"
```

```
FRONTEND_DOMAIN="https://app.example.com"
```

```
FRONTEND_DOMAIN_COOKIE=".example.com"
```

See here in "frontend domain" value there is the domain of your Frontend where you will host it. Ex.: app.johnsrestro.com

And in the front-end domain the value starts with "." (dot) and then the domain

***Make sure you configure these details carefully, failing to do so, the app will not work.***



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x24
GNU nano 7.2 backend/.env *
DATABASE_URL='mysql://root:MYSECRET@127.0.0.1:3306/restropro'
JWT_SECRET=restro_jwt_secret
JWT_EXPIRY=15m
JWT_EXPIRY_REFRESH=30d

# provide in ms - milliseconds => 15 minute, 30days | match the value to JWT expiry
COOKIE_EXPIRY=300000 # 15 minute
COOKIE_EXPIRY_REFRESH=259200000 # 1 day

PASSWORD_SALT=10
FRONTEND_DOMAIN="https://demorestro.uiflow.in"
FRONTEND_DOMAIN_COOKIE=".uiflow.in"

NODE_ENV=production

File Name to Write: backend/.env
^G Help          M-D DOS Format   M-A Append       M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend      ^T Browse
```

Once you edit the details, to exit the editor press Ctrl + X, then press Y, and save.

Now to start the backend server run the following command.

```
> pm2 start index.js
```



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend# pm2 start index.js
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~/backend -- ssh root@143.110.240.135 -- 93x24
$ pm2 start api.js -i 4

Monitor in production:
$ pm2 monitor

Make pm2 auto-boot at server restart:
$ pm2 startup

To go further checkout:
http://pm2.io/

-----

[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /root/backend/index.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	u	status	cpu	memory
0	index	fork	0	online	0%	14.0mb

```
root@ubuntu-s-1vcpu-1gb-blr1-01:~/backend#
```

That's it, your backend server is started on your Server's IP. But we've to link the domain too. For that we've to install and configure the nginx. So let's do it.

- > sudo apt update
- > sudo apt install nginx

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo apt update
sudo apt install nginx
```

Now we will enable Firewall, run the following commands.

- > sudo ufw app list
- > sudo ufw allow 'Nginx HTTP'
- > sudo ufw allow 'Nginx HTTPS'
- > sudo ufw allow ssh
- > sudo ufw enable
- > sudo ufw status



```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw status
Status: inactive
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw status
Status: active

To Action From
--
Nginx HTTP ALLOW Anywhere
Nginx HTTP (v6) ALLOW Anywhere (v6)

root@ubuntu-s-1vcpu-1gb-blr1-01:~# |
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ufw allow 'Nginx HTTPS'
```

After you allow rules in Firewall, we will need to add domain config and server config in nginx config, just follow the commands.

- > cd /etc/nginx/sites-available/
- > sudo nano demobackend.mydomain.com

Here look after the nano in the second command, you should put the domain of your backend, Ex: api.johnsrestro.com, the domain should be the same that we added in DNS config for Backend.



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: /etc/nginx/sites-available — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# cd /etc/nginx/sites-available/
root@ubuntu-s-1vcpu-1gb-blr1-01:/etc/nginx/sites-available# ls
default
root@ubuntu-s-1vcpu-1gb-blr1-01:/etc/nginx/sites-available# |
```



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: /etc/nginx/sites-available — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:/etc/nginx/sites-available# sudo nano demobackend.uiflow.in
```

Once the editor is opened, put following text there, make sure to replace the highlighted values with your backend domain.

```

server {
    listen 80;
    listen [::]:80;

    root /var/www/demobackend.uiflow.in/html;

    index index.html index.htm index.nginx-debian.html;

    server_name demobackend.uiflow.in;

    location / {
        proxy_pass http://localhost: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;
    }
}

```

The screenshot shows a terminal window with the nano text editor open. The editor is editing the file `demobackend.uiflow.in`. The content of the file is the same as shown in the previous block. The terminal window title is `restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: /etc/nginx/sites-available -- ssh root@143.110.240.135 -- 93x24`. The nano editor status bar shows `GNU nano 7.2 demobackend.uiflow.in *`. The bottom of the window displays a list of nano editor shortcuts: `^G Help`, `^O Write Out`, `^W Where Is`, `^K Cut`, `^T Execute`, `^C Location`, `^X Exit`, `^R Read File`, `^_ Replace`, `^U Paste`, `^J Justify`, and `^_ Go To Line`.

Now save the file and exit the editor.

We'll copy this newly created file from sites-available to sites-enabled. Use following command.

```
> cp demobackend.uiflow.in ../sites-enabled/demobackend.uiflow.in
```

Make sure you replace the provided domain with your backend domain.

A terminal window screenshot showing the execution of the command `cp demobackend.uiflow.in ../sites-enabled/demobackend.uiflow.in`. The terminal title bar indicates the user is root at an Ubuntu server (ubuntu-s-1vcpu-1gb-blr1-01) in the directory /etc/nginx/sites-available, connected via SSH to IP 143.110.240.135. The command prompt shows the user is in the /etc/nginx/sites-available directory.

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: /etc/nginx/sites-available — ssh root@143.110.240.135 — 93x24  
...untu-s-1vcpu-1gb-blr1-01: /etc/nginx/sites-available — ssh root@143.110.240.135  
root@ubuntu-s-1vcpu-1gb-blr1-01:/etc/nginx/sites-available# cp demobackend.uiflow.in ../sites-enabled/demobackend.uiflow.in
```

Once you do it, run the following command to check that your applied changes are error free.

```
> sudo nginx -t
```

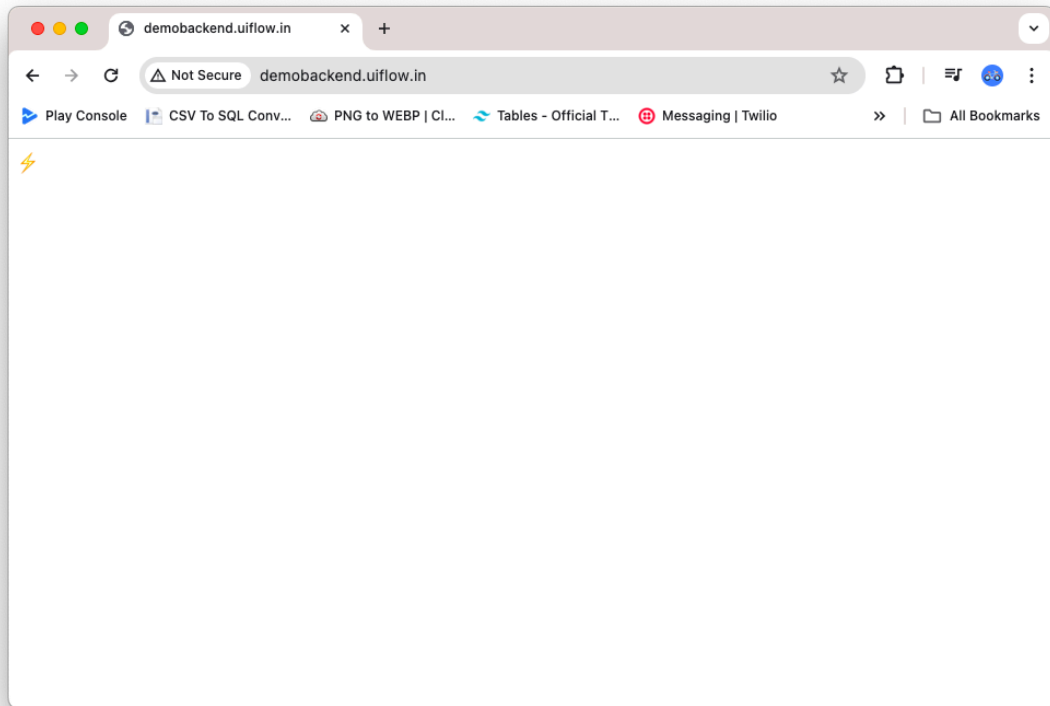
```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

If it says “test is successful” then it’s time to restart the nginx server.

```
> sudo systemctl restart nginx
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo systemctl restart nginx
```

After restarting the server, you can visit the backend domain on your browser.



Hooray 🙌 we've configured the domain for backend successfully! But still it's not Secured, since SSL installation is pending. Don't worry, it's very quick.

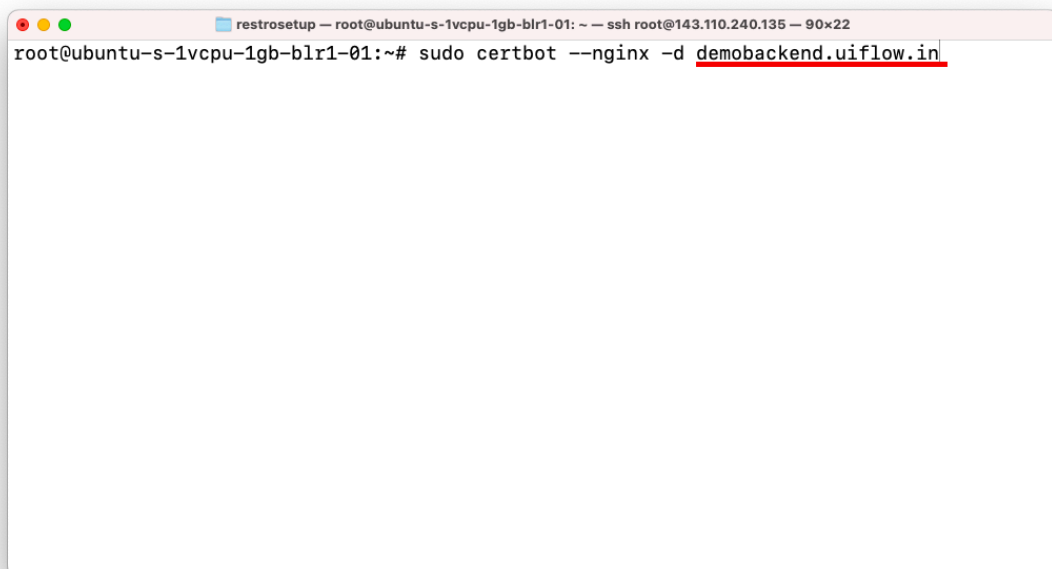
```
> sudo apt install certbot python3-certbot-nginx
```



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 93x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo apt install certbot python3-certbot-nginx
```

```
> sudo certbot --nginx -d demobackend.uiflow.in
```

Make sure you replace the domain with yours.



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 90x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo certbot --nginx -d demobackend.uiflow.in
```

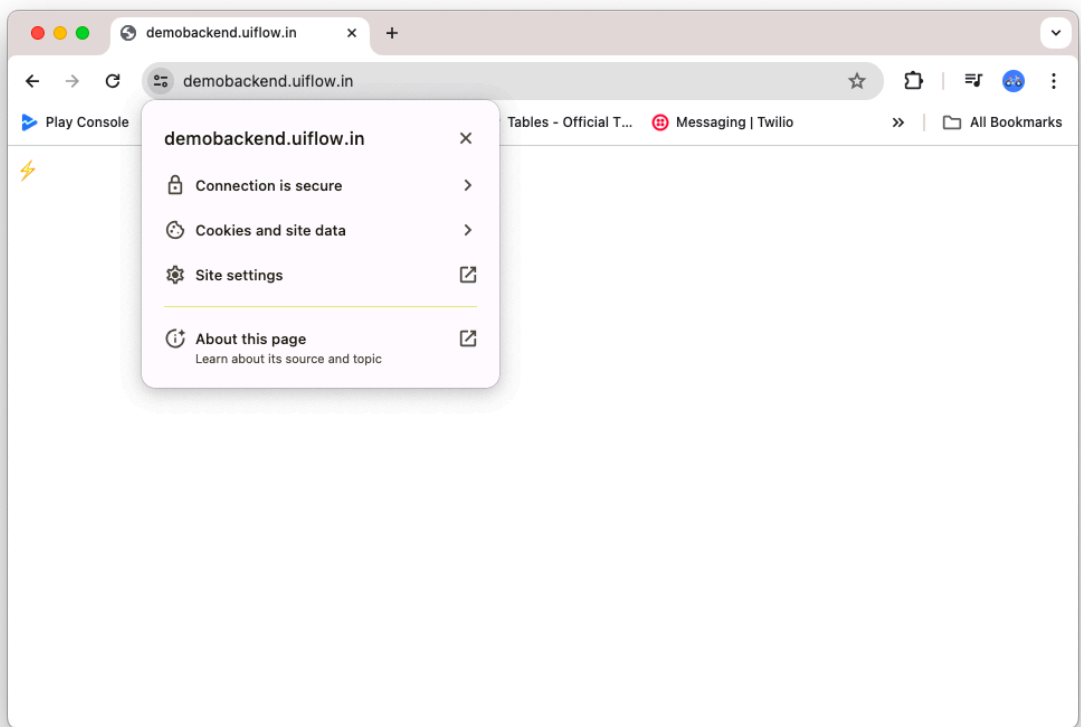
```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 90x22
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for demobackend.uiflow.in

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/demobackend.uiflow.in/fullchain.pem
Key is saved at: /etc/letsencrypt/live/demobackend.uiflow.in/privkey.pem
This certificate expires on 2024-07-12.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for demobackend.uiflow.in to /etc/nginx/sites-enabled/demobackend.uiflow.in
Congratulations! You have successfully enabled HTTPS on https://demobackend.uiflow.in

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

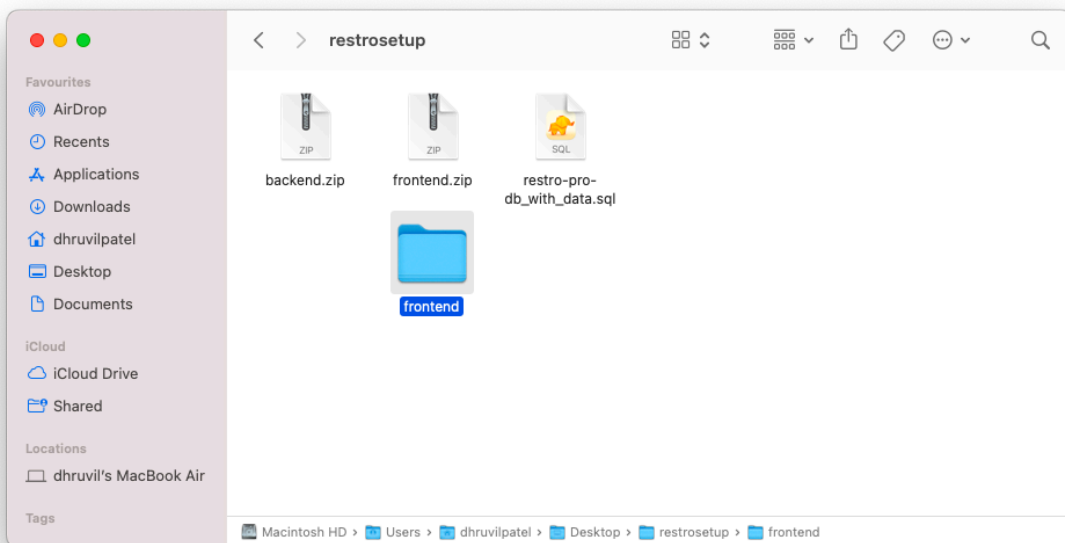
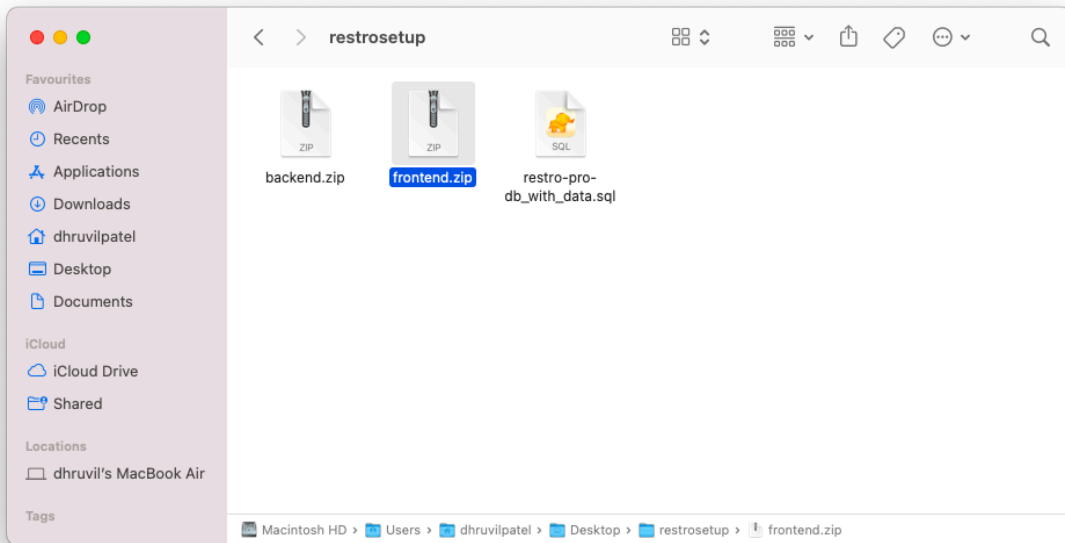
Now visit the backend domain URL again and refresh.



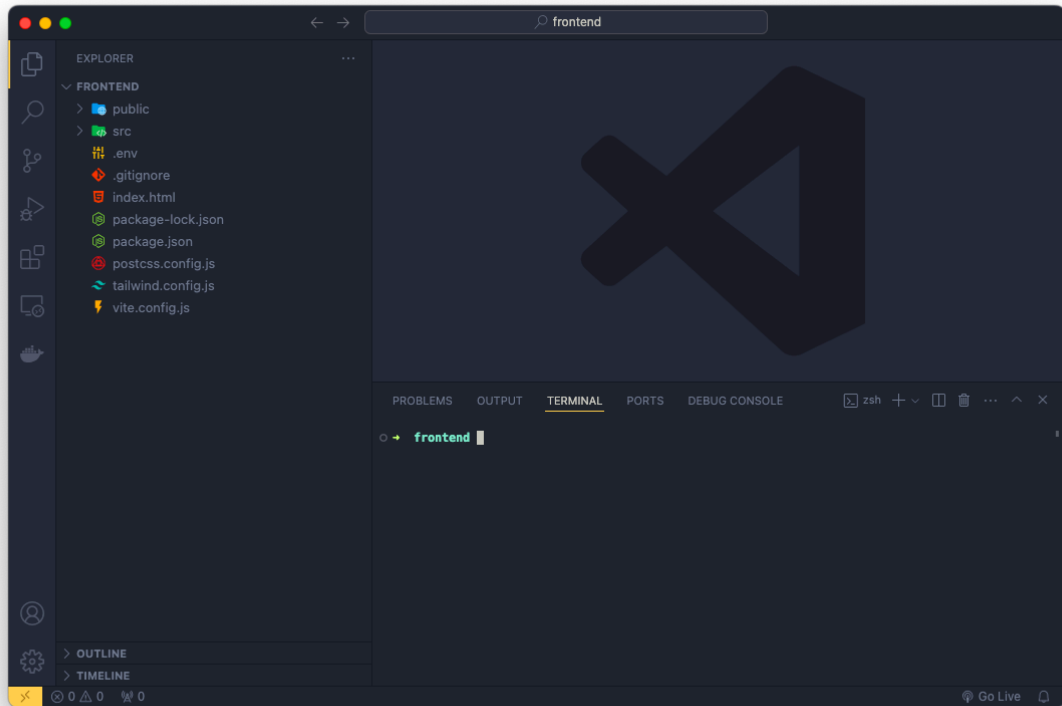
Yay! Our site is now HTTPS! 🕶️

## 5. Frontend code Deployment

Unzip the frontend code.

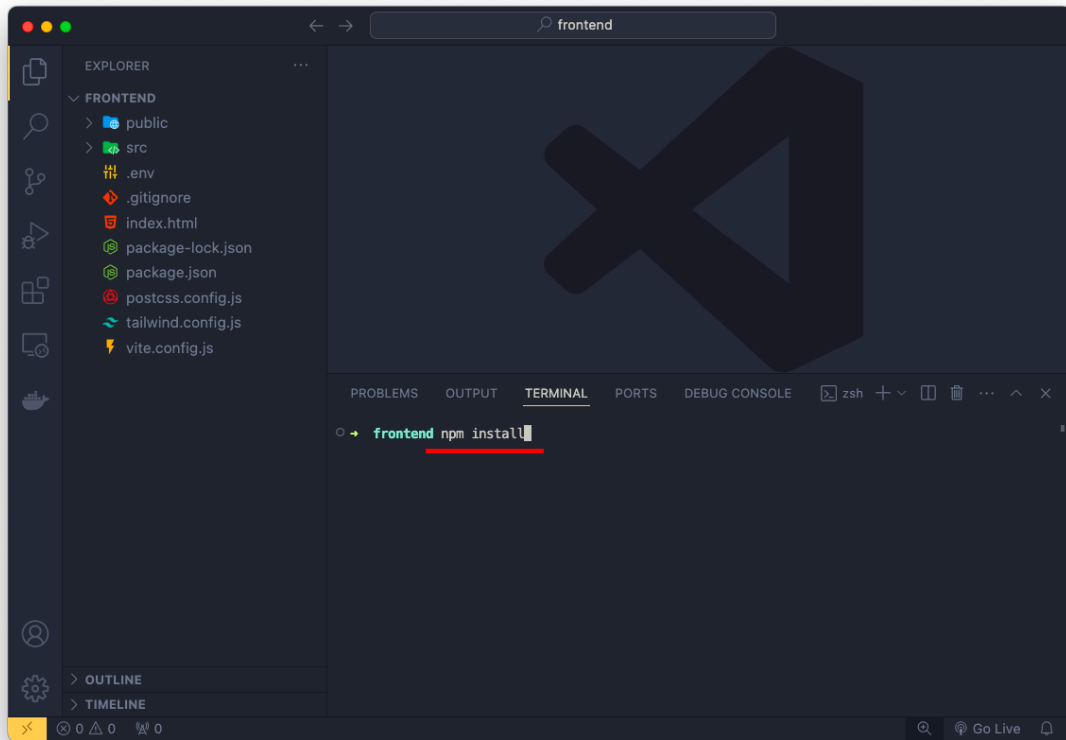


And open in the Code editor like VS Code.



Install frontend dependencies by running the following command in the frontend directory.

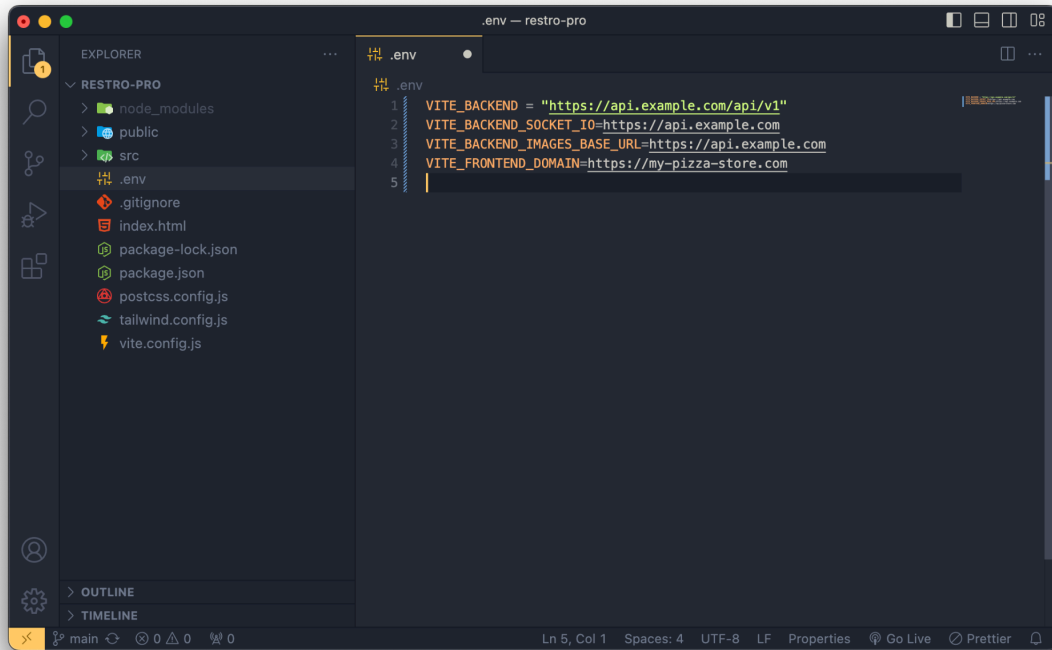
```
> npm install
```



Now open .env file, and replace the domain with your backend server domain.

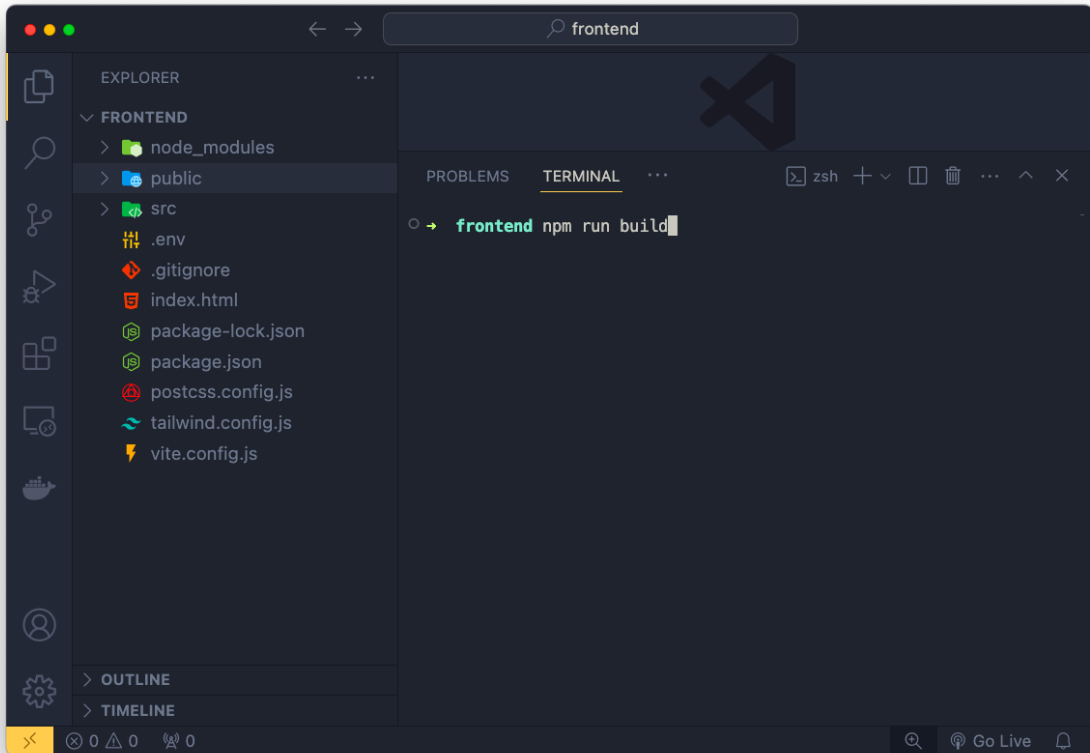
```
VITE_BACKEND="https://api.example.com/api/v1"  
VITE_BACKEND_SOCKET_IO="https://api.example.com"  
VITE_BACKEND_IMAGES_BASE_URL=https://api.example.com  
VITE_FRONTEND_DOMAIN=https://my-pizza-store.com
```

Note: There should not be any trailing slashes.

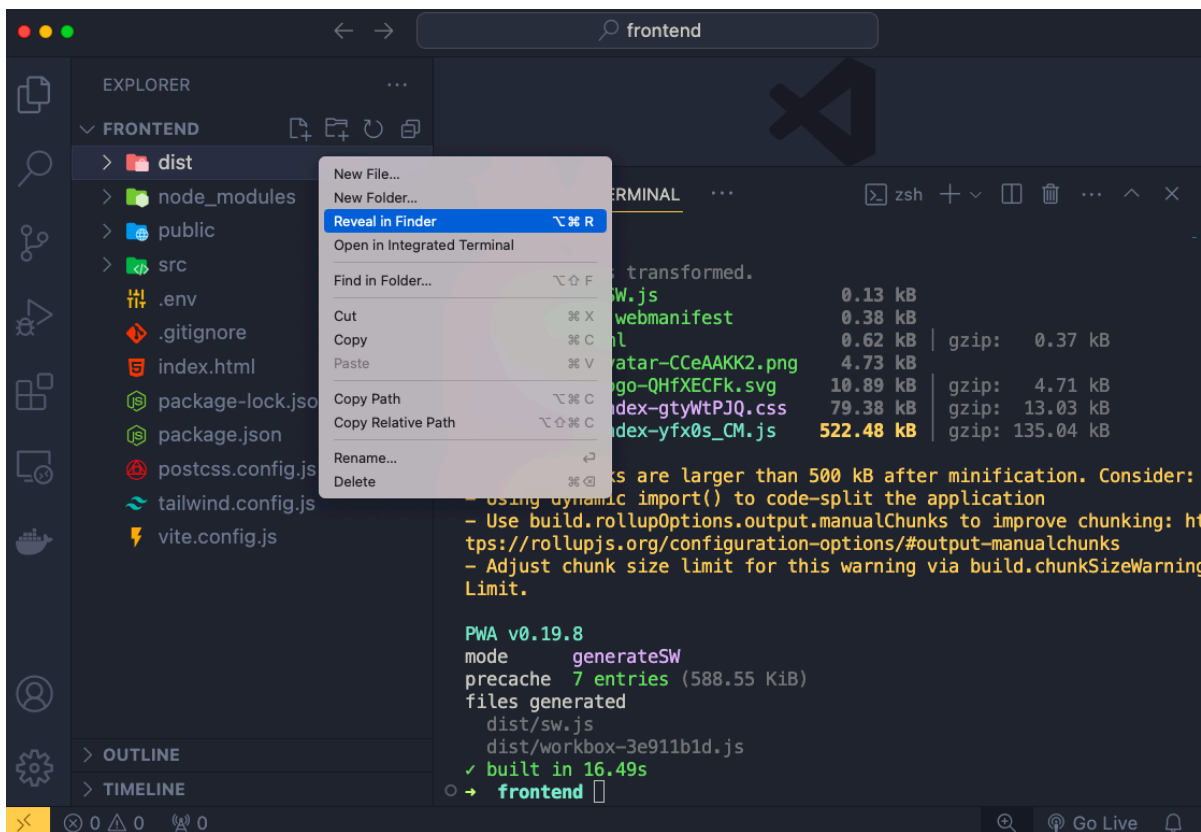


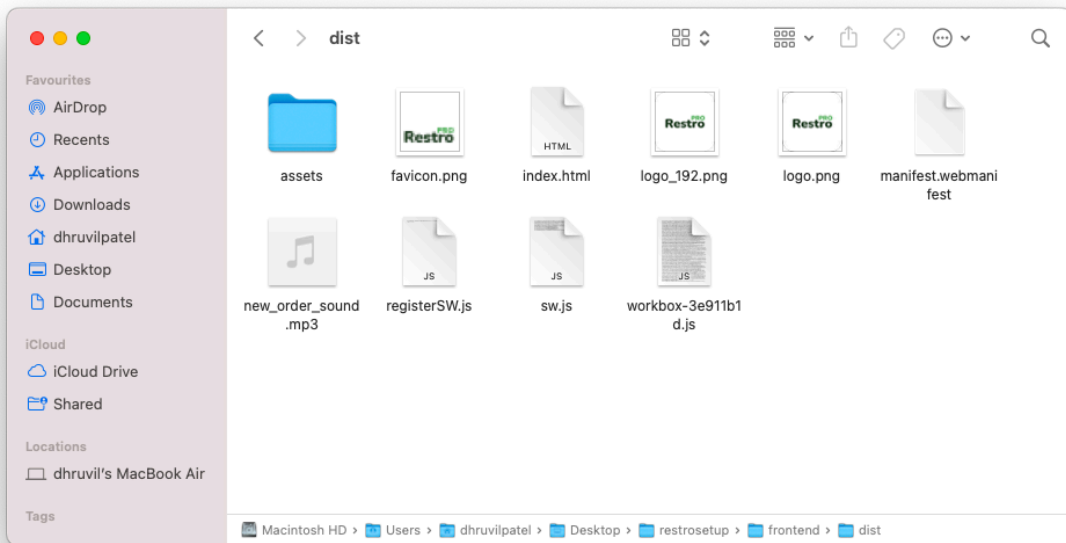
Now we will build the frontend code. Use following command.

```
> npm run build
```

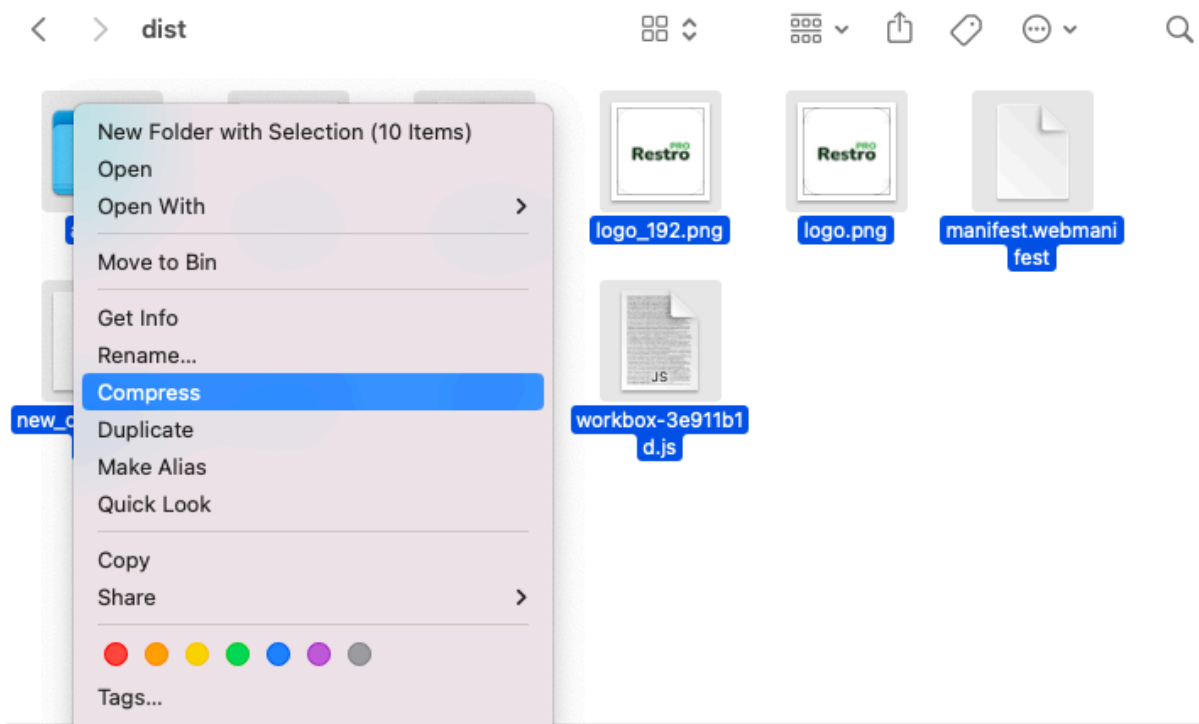


You'll see new folder appeared called "dist/" now open in Finder or File Explorer.





Select all files and make a zip file.



Now we will send this zip file to the Digital Ocean Droplet (Server).

```
restrosetup — scp restrobuild.zip root@143.110.240.135:~/ — scp — ssh · scp restrobuild.zip root@143.110.240.135:~/ — 97x22
→ restrosetup scp restrobuild.zip root@143.110.240.135:~/
```

Once send you can SSH into your server, and do quick “ls” to check.

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# ls
__MACOSX backend backend.zip restro-pro-db_with_data.sql restrobuild.zip snap
root@ubuntu-s-1vcpu-1gb-blr1-01:~# |
```

Now remember when editing DNS records in Backend Configuration section, if you haven't configured frontend domain that time then here is the time, let's do it.

Goto DNS Settings of your domain provider and add “A” Record.

Overview **DNS** Products

DNS Records Forwarding Nameservers Premium DNS Hostnames DNSSEC NEW

**DNS records** define how your domain behaves, like showing your website content and delivering your email.

**New Records**

[A records](#) use an IP address to connect your domain to a website. They're also used to [create subdomains](#) such as www or store, that point to an IP address.

Type *	Name *	Value *	TTL
A	demorestro	143.110.240.135	1/2 Hour

[+ Add another value](#)

Add More Records Save Cancel

Now we'll configure the NGINX server to serve frontend code.

```
> sudo mkdir -p /var/www/demorestro.uiflow.in/html
```

Note: Make sure you replace the frontend domain with your frontend domain.



```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# ls
__MACOSX backend backend.zip restro-pro-db_with_data.sql restrobuild.zip snap
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo mkdir -p /var/www/demorestro.uiflow.in/html
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

Grant the access to this newly created directory.

```
> sudo chown -R $USER: $USER /var/www/demorestro.uiflow.in/html
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo chown -R $USER:$USER /var/www/demorestro.uiflow.in/html
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

```
> sudo chmod -R 755 /var/www/demorestro.uiflow.in
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo chmod -R 755 /var/www/demorestro.uiflow.in
```

```
> sudo chmod -R 755 /var/www/demorestro.uiflow.in
```

A terminal window with a title bar that reads "restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x22". The terminal content shows the command "root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo chmod -R 755 /var/www/demorestro.uiflow.in" followed by a cursor at the end of the line.

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 97x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo chmod -R 755 /var/www/demorestro.uiflow.in
```

Now move the frontend zip file to this folder.

```
> mv restrobuild.zip /var/www/html/resrobuild.zip
```

A terminal window with a title bar that reads "restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 101x22". The terminal content shows the command "root@ubuntu-s-1vcpu-1gb-blr1-01:~# mv restrobuild.zip /var/www/html/resrobuild.zip" followed by a new prompt "root@ubuntu-s-1vcpu-1gb-blr1-01:~#".

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 101x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# mv restrobuild.zip /var/www/html/resrobuild.zip
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

Unzip the code.

```
> unzip restrobuild.zip
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html -- ssh root@143.110.240.135 -- 101x22
root@ubuntu-s-1vcpu-1gb-blr1-01:/var/www/html# ls
index.nginx-debian.html  resrobuild.zip
root@ubuntu-s-1vcpu-1gb-blr1-01:/var/www/html# unzip resrobuild.zip
Archive:  resrobuild.zip
  creating: assets/
  inflating: assets/index-yfx0s_CM.js
  inflating: assets/logo-QHfXECfk.svg
  inflating: assets/circle_illustration.svg
  inflating: assets/avatar-CCeAAKK2.png
  inflating: assets/index-gtyWtPJQ.css
  inflating: favicon.png
  inflating: index.html
  inflating: logo_192.png
  inflating: logo.png
  inflating: manifest.webmanifest
  inflating: new_order_sound.mp3
  inflating: registerSW.js
  inflating: sw.js
  inflating: workbox-3e911b1d.js
root@ubuntu-s-1vcpu-1gb-blr1-01:/var/www/html#
```

Now you can remove zip file.

```
> rm resrobuild.zip
> rm index.nginx-debian.html
```

```
restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html -- ssh root@143.110.240.135 -- 101x22
root@ubuntu-s-1vcpu-1gb-blr1-01:/var/www/html# rm ./resrobuild.zip
root@ubuntu-s-1vcpu-1gb-blr1-01:/var/www/html#
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html — ssh root@143.110.240.135 — 101x22
root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html# rm index.nginx-debian.html
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html — ssh root@143.110.240.135 — 101x22
root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html# rm index.nginx-debian.html
root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html# ls
assets          index.html     logo_192.png  new_order_sound.mp3  sw.js
favicon.png    logo.png      manifest.webmanifest  registerSW.js         workbox-3e911b1d.js
root@ubuntu-s-1vcpu-1gb-blr1-01: /var/www/html#
```

It's time to configure the NGINX settings to serve the frontend code to the frontend domain.

```
> sudo nano /etc/nginx/sites-available/demorestro.uiflow.in
```

Note: make sure you replace the domain with your frontend domain

A terminal window with a title bar that reads "restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 101x22". The terminal prompt is "root@ubuntu-s-1vcpu-1gb-blr1-01:~#". The command "sudo nano /etc/nginx/sites-available/demorestro.uiflow.in" has been entered and is currently at the end of the line.

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 101x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo nano /etc/nginx/sites-available/demorestro.uiflow.in
```

And paste the following text.

```
server {
  listen 80;
  listen [::]:80;

  root /var/www/demorestro.uiflow.in/html;

  index index.html index.htm index.nginx-debian.html;
  server_name demorestro.uiflow.in;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

```
GNU nano 7.2 /etc/nginx/sites-available/demorestro.uiflow.in
server {
  listen 80;
  listen [::]:80;

  root /var/www/demorestro.uiflow.in/html;
  index index.html index.htm index.nginx-debian.html;

  server_name demorestro.uiflow.in;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

Save the file and exit editor.

Now copy the file from sites-available to sites-enabled

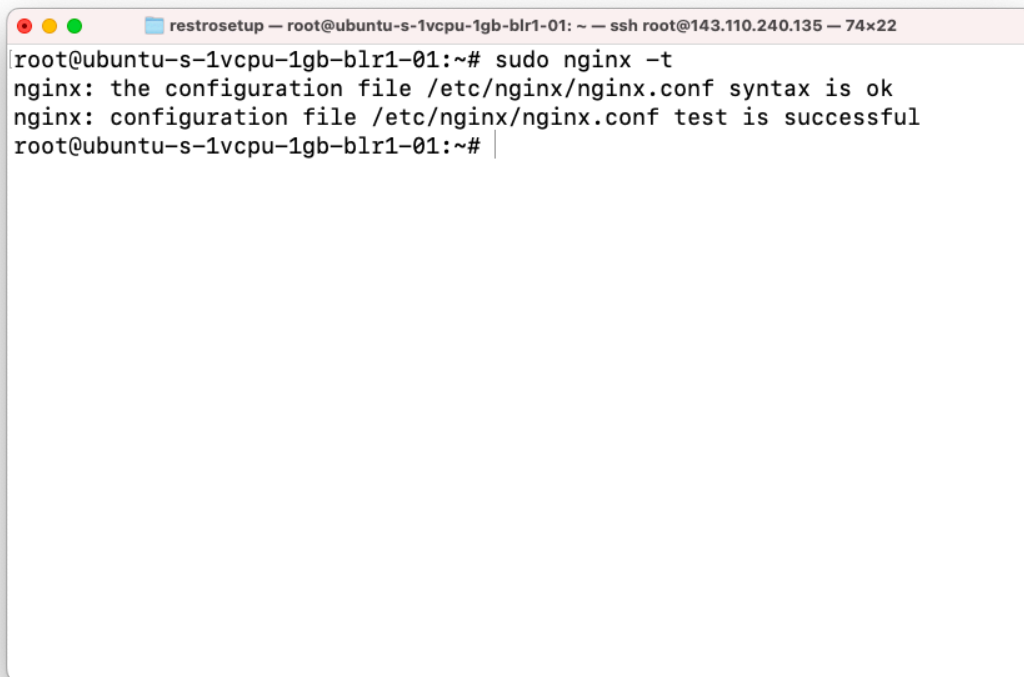
```
> sudo ln -s /etc/nginx/sites-available/demorestro.uiflow.in /etc/nginx/sites-enabled/
```

Note: Make sure you replace the domain.

```
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo ln -s /etc/nginx/sites-available/demorestro.uiflow.in /etc/nginx/sites-enabled/
```

Check that your applied changes are correct or not, by running following command.

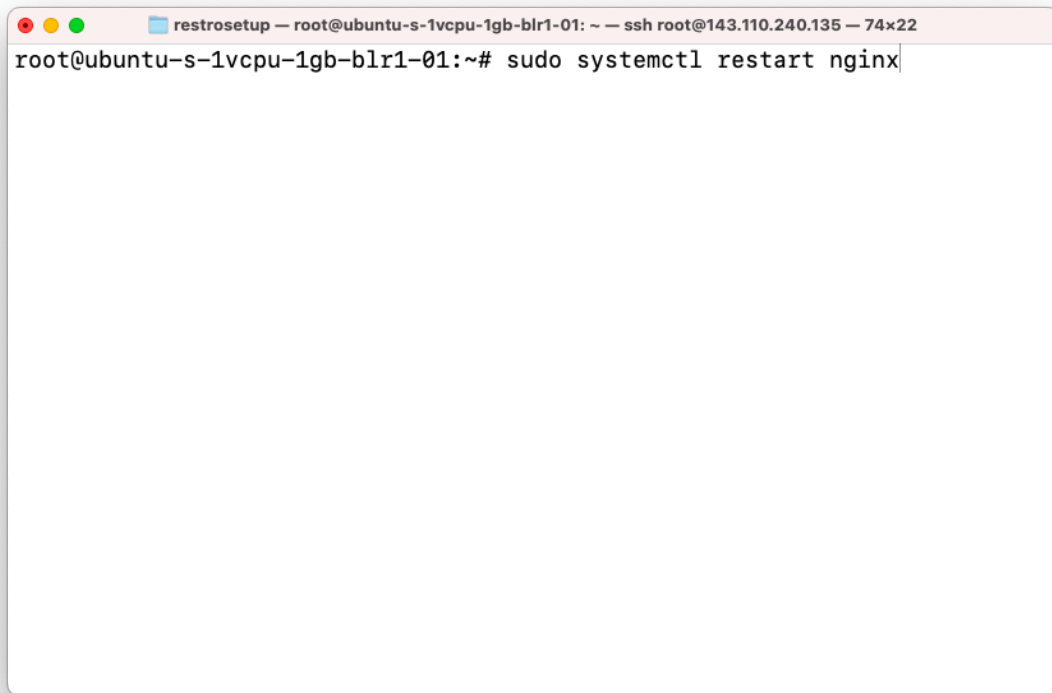
```
> sudo nginx -t
```

A terminal window with a title bar that reads "restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 74x22". The terminal content shows the command "sudo nginx -t" being executed, resulting in two lines of output: "nginx: the configuration file /etc/nginx/nginx.conf syntax is ok" and "nginx: configuration file /etc/nginx/nginx.conf test is successful". The prompt "root@ubuntu-s-1vcpu-1gb-blr1-01:~#" is visible at the end of the second line.

```
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```

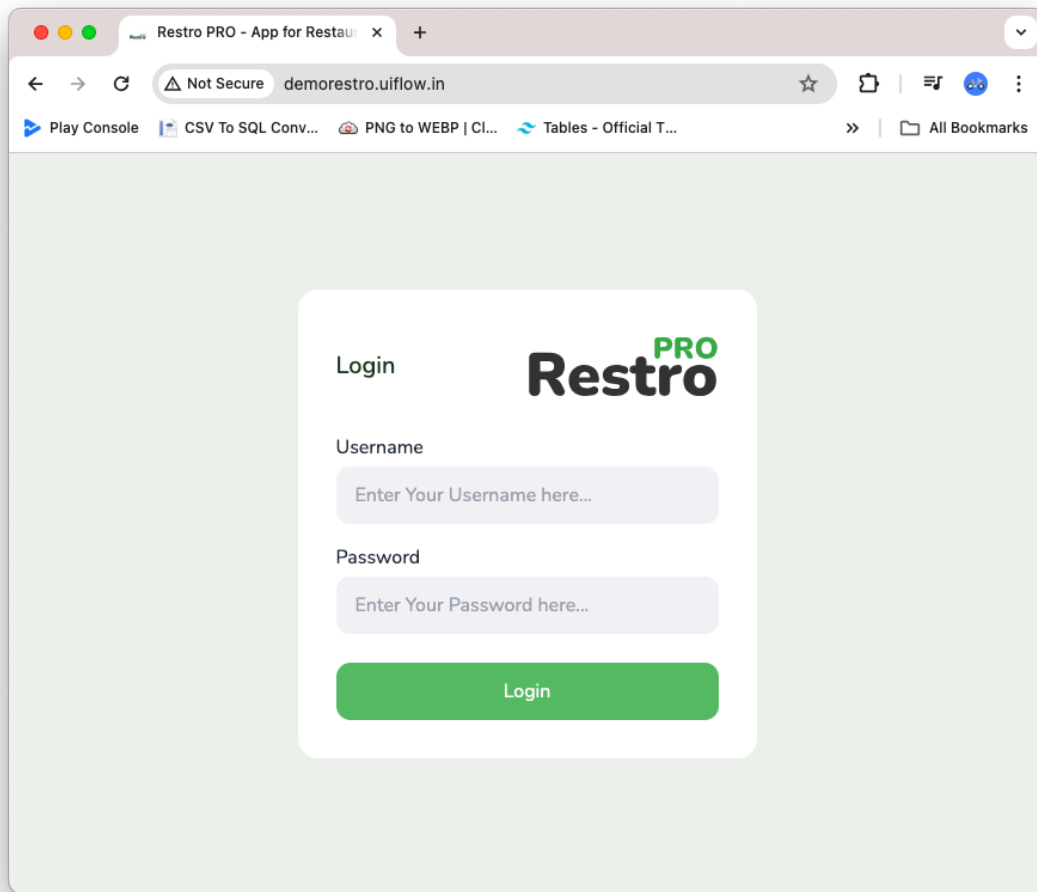
Now restart the nginx server.

```
> sudo systemctl restart nginx
```

A terminal window with a title bar that reads "restrosetup -- root@ubuntu-s-1vcpu-1gb-blr1-01: ~ -- ssh root@143.110.240.135 -- 74x22". The terminal content shows the prompt "root@ubuntu-s-1vcpu-1gb-blr1-01:~#" followed by the command "sudo systemctl restart nginx" with a cursor at the end of the line.

```
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo systemctl restart nginx
```

Now you can visit the frontend domain, and you should see the app running.



Hooray 🎉 only thing remaining is to install SSL for frontend domain.

```
> sudo certbot --nginx -d demorestro.uiflow.in
```

Note: make sure you put your frontend domain.

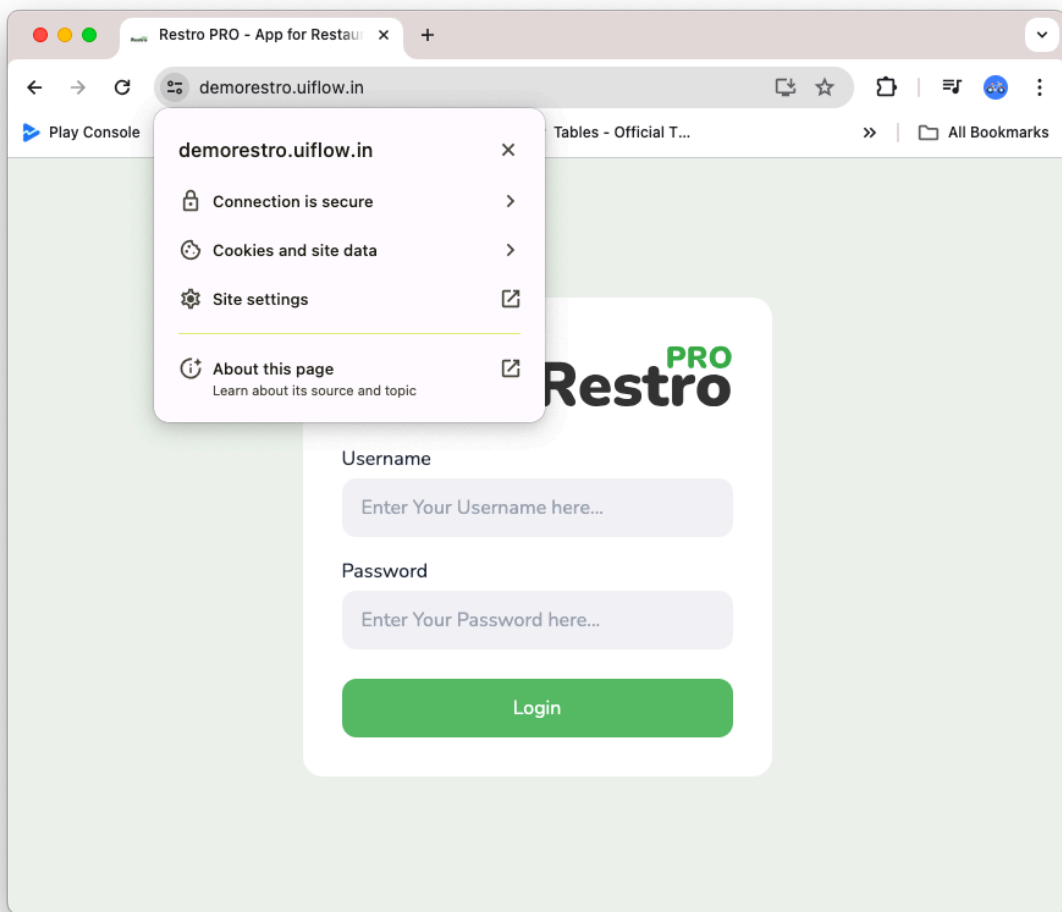
```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 91x22
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo certbot --nginx -d demorestro.uiflow.in
```

```
restrosetup — root@ubuntu-s-1vcpu-1gb-blr1-01: ~ — ssh root@143.110.240.135 — 91x24
root@ubuntu-s-1vcpu-1gb-blr1-01:~# sudo certbot --nginx -d demorestro.uiflow.in
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for demorestro.uiflow.in

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/demorestro.uiflow.in/fullchain.pem
Key is saved at: /etc/letsencrypt/live/demorestro.uiflow.in/privkey.pem
This certificate expires on 2024-07-12.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for demorestro.uiflow.in to /etc/nginx/sites-enabled/demorestro.uiflow.in
Congratulations! You have successfully enabled HTTPS on https://demorestro.uiflow.in

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
root@ubuntu-s-1vcpu-1gb-blr1-01:~#
```



Yay! App is now HTTPS SSL.

## 6. Configuring Settings of RestroPRO

### 6.1: Demo and App Settings

<https://www.youtube.com/watch?v=WuSNCqsSnk0>

### 6.2: Creating Super Admin User

Open <https://bcrypt-generator.com/> and paste your plain text password in the field and use "10" rounds. And click on the Encrypt button. Now copy the encrypted text and paste it somewhere temporary.

# Encrypt

Encrypt some text. The result shown will be a Bcrypt encrypted hash.

```
$2a$10$ivrE8krMBMkRIZXw6JPRQeegdnn36.0WChlxJJobP1mEtnM8Dj9H2
```

Rounds

Login to your server and open up mysql cli. And run the following query. Make sure to replace the placeholder values;

```
INSERT INTO `superadmins` VALUES  
(`example@gmail.com`,`paste_here_your_encrypted_password`,`Super  
Admin user name`);
```

## 7. Our Installation & Customisation Service

We understand that setting up a Node.js and React project on a server can be a technical task. That's why we offer a deployment service to take care of everything for you. This service is perfect for those who are not comfortable with the technical aspects or simply want to save time.

Contact us on: [hi@uiflow.in](mailto:hi@uiflow.in)

**Need  
Installation  
Service?**

✉ [hi@uiflow.in](mailto:hi@uiflow.in)

**+69\$  
only**

**Looking for customisation?** You're welcome Contact us on: [hi@uiflow.in](mailto:hi@uiflow.in)

Looking for a Development Outsourcing Service?

We offer custom app development services. Our team of experienced developers & designers will work closely with you to understand your specific needs and build a custom application that perfectly fits your workflow.

Contact us on: [hi@uiflow.in](mailto:hi@uiflow.in)

# ***Crafting the Perfect Software. We Build it for You.***

Our software empowers businesses. This POS system is a prime example, but we don't stop there. ***Need a custom solution?*** Contact us today - our team will craft a proposal within 24 hours.

✉ [hi@uiflow.in](mailto:hi@uiflow.in)