INTERVIEW TASK COMPUTAN DOCUMENTATION

VT Documentation

1. Prerequisite
2. Create Authentication For Admins and User
   1. Database
   2. Middleware Authentication
   3. Login
3. Import CSV records into Database
   1. upload csv file
   2. Dump all records into database
4. Read data From an API and populate the database
5. Exception Handling
6. Create Jobs To Email all active users

VT Documentation

1. Prerequisite

Entry Point : “/all”

Database Configuration: You need to configure the database in the .env file in Comptan-Interview-Task folder

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=computan

DB\_USERNAME=root

DB\_PASSWORD=

Mail Configuration: You need to setup your host, port, username and password in .env file in Comptan-Interview-Task folder

MAIL\_MAILER=smtp

MAIL\_HOST=smtp.mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=\*\*\*\*\*\*\*\*

MAIL\_PASSWORD=\*\*\*\*\*\*\*\*

MAIL\_ENCRYPTION=tls

MAIL\_FROM\_ADDRESS="hello@example.com"

MAIL\_FROM\_NAME="${APP\_NAME}"

Migrate Database: Run PHP “PHP artisan migrate:refresh –seed” on the terminal inside Comptan-Interview-Task directory

laravel-vincent-tosin\computan-interview-task> php artisan migrate:refresh --seed

VT Documentation

1. Create Authentication For Admin and User

The dedication url for Admin login page is “/admin”

The dedication url for User login page is “/”

The home routes are authorized by auth middleware, a custom middleware I created, register under that name in app/http/kernel.php

* 1. Database

There is a separate database table of admins for admin(coloumn: id, email, username and password) and of users for User(coloumn: id, email, username and password,is\_active)

* 1. Middleware Authorization:

After authentication in AuthContoller- auth method a session variable (‘role’) is set for;

**Users:** Login route **‘/’** , role=**user**,

If any request is made to **“/home”** route, and **session(‘role’)** doesn’t exist or the value **session(‘role’)** is not **user** it automatically redirects to **‘/’** for Login.

**Admins:** Login route **‘/admin’** , **role=admin**,

If any request is made to **“/admin\_home”** route, and **session(‘role’)** doesn’t exist or the value session(‘role’) is not **admin** it automatically redirects to ‘/’ for Login.

* 1. LOGIN
     1. Checks if the request to Login is from a User or an Admin
     2. Validate user credentials with the user table and redirect to /home
     3. Or validate Admin credential with Admin table and redirect to /admin\_home

User and Admin can Login with **username and password** or **email and password**

VT Documentation

1. **Import CSV records into database**

The dedicated Url for this service is “/home” can only be accessed by authorized users

“/home”: supports the route get method, for users to select a file and submit, note only csv file is allowed, it is submitted to

“/csvupload”: supports route post method

1. Checks if It is a csv file and if the request have a csv file
2. Convert CSv file to an array
3. Chunk it to 300 records for Queue worker to process each Chunk in batch using the Csv Processor Job.
4. Each chunk is inserted into database
5. Each record is unique by ID
6. CSV coloumns must be

* Any identifier(ID, record\_id or whatsoever)
* Name
* Email
* Phone
* Address

1. Read data From an API and populate the database

The dedicated Url for this service is “/read-dfa”

Only accept Json, if request fails, it redirects to 404 error page and send mail to ‘tech@computan.com’

1. Each record is unique by API
2. And are stored in the entry table using the EntryProcessor Job.
3. Exception Handling
4. If page doesn’t exist, user are redirected to 404 error page
5. In case of any fatal error user are routed to 404 error page
6. In case of an exception [tech@computan.com](mailto:tech@computan.com) receive an email.
7. Create Jobs To Email all active users
8. Email is sent to every user whose is\_active value is 1 on the user database table using a Mailer Job in Mailer.php in App/Jobs folder.

Url: “/mailer” – get - returns view to enter subject and body, submits to “/mailer” route (that supports post method)