**Name** : Suraj Gavhane

**Assignment** :

Exchanger

This application will show exchange rates for different currencies against USD as base currency.

1. **Installation:**

a. Git clone https://github.com/surajga/exchanger-app.git exchanger-app

b. cd exchanger-app

c. Run command: composer install

**2. Configurations and migrations**

\* Update DB configs in DATABASE\_URL property in .env

a. Create a new DB:

php bin/console doctrine:database:create #Create Database: exchange\_rates

b. Run Migration:

php bin/console doctrine:migrations:migrate

**3. Unit Testing**

Run below command

1.Unit test for controller

php bin/phpunit tests/Controller/ExchangeRateControllerTest.php

2.Unit test for service

php bin/phpunit tests/Controller/ExchangeRateControllerTest.php

**4. To run the application follow instructions given at** https://symfony.com/doc/current/setup.html#running-symfony-applications

[use php bin/console server:start \*:port if php bin/console server:start does not work]

**5. Navigations (By default URL will be** <http://127.0.0.1:8000>**)**

<http://127.0.0.1:8000> will be the index page showing up dashboard to view exchange rates hyperlink.

Hyperlink will lead to new page having all details list of exchange rates those are available in database table

http://127.0.0.1:8000/rates

**6. Completed Functionality**

1.Fetch Values exchange rate values from API service and store them into database table.

http://api.exchangeratesapi.io/

2.Fetch stored values from database table display on web page.

3.Add new exchange rates using UI interface.

4.Update existing exchange rate using UI interface.

5.Delete existing exchange rate using UI interface.

6.Unit test cases created for

a. API request and response (refresh exchange rate)

b. Add new exchange rate

c. Update existing exchange rate

d. Delete exchange rate

**7.Other**

1. NetBeans IDE has been used throughout the development.

2. PHPUnits are not yet complete. Unit test created only for Add, Edit, Delete exchange rate and API service to refresh exchange rate.

3. The code was reviewed (at a very high level ) once by an architect.