***Coding exercise:***

Write a Flask/Django App with only 1 method called “ProcessPayment” that receives a request like this

- CreditCardNumber (mandatory, string, it should be a valid credit card number) - CardHolder: (mandatory, string)

- ExpirationDate (mandatory, DateTime, it cannot be in the past)

- SecurityCode (optional, string, 3 digits)

- Amount (mandatoy decimal, positive amount)

The response of this method should be 1 of the followings based on

- Payment is processed: 200 OK

- The request is invalid: 400 bad request

- Any error: 500 internal server error

The payment could be processed using different payment providers (external services) called: - PremiumPaymentGateway,

- ExpensivePaymentGatewa or

- CheapPaymentGateway.

The payment gateway that should be used to process each payment follows the next set of business rules: a) If the amount to be paid is less than £20, use CheapPaymentGateway.

b) If the amount to be paid is £21-500, use ExpensivePaymentGateway if available. Otherwise, retry  only once with CheapPaymentGateway.

c) If the amount is > £500, try only PremiumPaymentGateway and retry up to 3 times in case  payment does not get processed.

**Optional**

Write a method “EstimatePrice” in your Flask / Django app that receives a date and then outputs the value  of the estimated stock price. The “Estimated Stock Price” is defined as the mid-range price between the  “High” and the “Low” price.

To predict the value of the estimated stock price, use the data set below. You can use any learning or  predictive algorithm for this method.

http://archive.ics.uci.edu/ml/machine-learning-databases/00312/

The response of this method should be one of the following based on

- Request is successful and value is returned: 200 OK

- The request is invalid: 400 bad request

- Any other exception / error: 500 internal server error

*Recommendations*:

- The classes should be written in such way that they are easy to test.

- Write as many tests as you think is enough to be certain about your solution works - Use SOLID principles.

- Decouple the logic the prediction logic from the API as much as possible