Q) Why do we want to use negative sampling and stochastic gradient descent together in a skip-gram model?

There is a very subtle point you are missing. There are two things due to which the training cost increases:

1. Calculation of denominator: As you know, the calculation of denominator involves summing over all the training examples, which is expensive. Negative sampling solves this problem. For every (context, target) pair, we randomly sample some words whose label we want to be 0, called the negative samples.
2. Calculation of error: Notice that you need to pass through the whole data corpus, and calculate the loss taking every word as a context. I am talking about the summation you do over the entire corpus. This is also very expensive. Hence what we do is that we only take one word as a context, calculate the loss for that word as a context(where we can use negative sampling to help is in calculating the giant denominator), and update the weight vectors then only. In the standard case, we would take the sum of all such losses, and then only we would have updated the weights.

Now, the point is that even if we are taking a few or one context word for calculating the loss and making the update(thus tackling problem 2), we still don't want to calculate the giant denominator. It is essential to understand that the denominator is independent of whether we use stochastic descent or not. Stochastic descent changes the number of loss terms we sum, before making the update. The loss term essentially remains the same.

Q2) What is this j ~ P(w) in the negative sampling formula?

It just means that j is sampled from the distribution P(w), which is that unigram model to the power 3/4. It refers to the j negative sample words.