Choice of technology stack

For our project, our team decided to utilize Django. Most of us are unfamiliar with creating an application from scratch, but have experience with coding in Python. While we're willing to learn new languages for our project, we don't want to start at square one, spending more time on learning new programming languages instead of actually working on implementing the features of the project. Our choice of Django (over other python-based stacks like Flask) is also due to the fact that one of our team members is familiar with Django. A bit of research shows that the Django community is bigger than the Flask community; finding information when we are stuck on our project would be easier to come by.

Additionally, Django handles many common tasks that are a significant part of any web application. It has HTML templates that we can use to build a response for our user without writing a lot of code ourselves. This lets us focus on the creative parts of our application while letting Django handle a lot of the production side. Django includes a lot of ready-made solutions in comparison to Flask, helping us build our application faster. Django's standardized and simplified development process is easier for beginners like us to use, while Flask's flexibility might prove to be a roadblock for us because we are unsure of what to do.