Tool	**Benefits**	**Downsides**
PyCharm Debugger	• Easy to set up • Easy to integrate with Docker/Flask • Visual Debugging	• In order to debug Flask commands you need to create a specific running configuration for each command and particular command line arguments
Jupyter	• Visual debugging • Easy way to check variable values quickly • Can run independent code fragments	• It's a hustle to set up and run a whole Flask project
PuDB	• Visual debugging	• Better suited for single file debugging • Hard to integrate with Docker/Flask
pdb	• Easy to use • Quick • Works with Docker	• Command line based debugger

After investigating the options above for debugging the project, I think we should settle on the PyCharm debugger, since it is the easiest visual debugging tool to setup and use. The disadvantage of other tools is that they are significantly difficult to integrate with Docker and/or Flask. As a fallback for quick debugging we can use the `pdb` module.

Steps to setup the project for debugging in PyCharm:

1. Create a new Docker Compose interpreter and select `ao` as the service. You may need to create a Docker server (for this you can use the default settings)

2. Create a new run configuration. It should be a Flask server. Select the previously created Docker Compose interpreter as the Python interpreter and set `-host=0.0.0.0` as additional options and `FLASK_APP=reordering.app.create_app;FLASK_DEBUG=1` as Environment variables.

3. You can now set breakpoints within the code and run the project in debug mode (by clicking the debug button in the right upper corner)

Debugging Flask commands with the PyCharm debugger:

The previous setup does not allow for debugging commands. In order to do that, a new run configuration is required for each command.

Steps to debug Flask commands:

1. Create a new Python configuration and select the Module name instead of the Script path as the target to run. Set the Module name to *flask. Make sure the Python interpreter is the Docker Compose interpreter defined previously*

2. Set the command you want to run complete with command line arguments in the Parameters filed. Also add `FLASK_APP=reordering.app:create_app` and `FLASK_ENV=development` as additional Environment variables

3. Now you can set breakpoints in the code of the commands and run them in debug mode.