

# Acceptance Criteria

## General:

- All tasks associated with a story must be completed according to the Definition of Done.
- Completion of a user story must be made known to fellow team members.
- Design docs should be updated as appropriate on user story completion.

## Specific to User Story:

- As a developer, I want to be able to scrape data from a set of job postings.
  - Scraper should be able to scrape at least 4 full pages of job data with no major errors.
  - Scraper should be able to run with minimal errors, though eliminating errors completely in this environment may not be feasible.
  - If the scraper does encounter an error, it should be handled in the code, if possible.
  - Scraper should not scrape duplicate.
- As a developer, I would like formatted data to be uploaded to the database
  - Data should be validated before attempting to upload it to the database.
  - An error message should be displayed when invalid data is given to be uploaded to the database.
  - Fields with no value in a particular job post should appear in the database as NULL
  - If the data needs to be put into a table that does not exist then a new table should be created
- As a developer, I want to have that data be formatted in a way that can interface with a database
  - Each subcategory of the formatted data should be contained in its appropriate data structure
  - The data formatter should be able to extract the required fields from a job post with minimal errors
- As a user, I want to be able to have a Webapp to access the calculations on the scraped data
  - Clicking add button should create a new empty card
  - Selecting the job title should display a pie chart with (data -- change later)
  - Clicking skills / languages should change the pie chart with different values