**Mighty**

**Prompt**

Claire is a budding entrepreneur, and she thinks she has the next great idea: an online vinyl records rental service. In Claire's service, customers will have a queue of records they want to rent (think old-school Netflix). Despite having an awesome idea, Claire wants to validate her business plan before proceeding full steam ahead. She wants to see if people are even willing to rent records online. Your task: build a web app for Claire’s online record shop idea.

**Requirements**

* Customers should be able to browse our catalog of records.
* Each customer should have their own queue they can interact with — they should be able to add, remove, and change the order of as many records as they want in their queue.
* Store administrators should be able to manage inventory (add, delete, and edit records) and keep track of stock.

**Assumptions**

* Customers can have up to 3 records rented out at a time, but as many queued up as they want.
* The record store can have multiple copies of the same record in stock.
* A single customer cannot rent out more than one copy of the same record at a time.
* If a record is out of stock, it can still be added to the queue, but it won’t be sent out until the next time it is in stock.

**Technology**

* Build Claire’s web app using Django. You may find the [Django tutorial](https://docs.djangoproject.com/en/3.2/intro/tutorial01/) useful.
* Use a relational database (Django has great support for them out of the box).
* You don't necessarily need to add any front end component, but we should be able to hit endpoints that fulfill these requirements
* Keep your code in a publicly accessible Git repository and link to it.
* Deploy a working version of this somewhere that is publicly accessible and link to it (e.g. Heroku or DigitalOcean).
* Write tests where you see fit.

**Pro Tips**

* Add a requirements.txt file for your project.
* You can leverage Django’s built-in admin for the administrative site.
* Document any trade-offs or work-arounds you made.