

# BeerVector - BluVector Exercise

---

This is an web application that allows visitors add, rate and leave reviews for beers

## Set Up / Running App

---

- Download & unpack .zip file
- Assuming you already have *python* and *pip*, install requirements using `pip install -r requirements.txt` within command prompt or terminal (*if not, check out [How to install python & pip](#)*)
- Create an admin user with the command `python manage.py createsuperuser`
- Run the command `python manage.py runserver`
- On your browser, go to <http://127.0.0.1:8000/> or whatever server address was provided in terminal

*Note: On your browser, go to <http://127.0.0.1:8000/admin> to access admin*

## Exercise Requirements

---

- Use a Python framework of your choice for the server side processing
- Use a persistent database of your choice to store the data
- Use a front-end design/layout framework of your choice to provide the base styling and structure for the application
- Utilize AJAX to populate the data on the landing

## How App Works (Expectations)

---

### View Product

Upon visiting the site, you shall be presented a view of the highest rated items, as well as a view of the most recently rated/reviewed items. This page also shows the name, number of reviews and overall rating for each beer. *This page includes a heading as well as all featured breweries added by admin*

### View Product Details

Navigating to a beer returns a detailed view of it (i.e. beer name, brewery, style, ABV, description & overall rating) as well as any reviews (rating, username of reviewer, time of comment & comments left) that have been written for it.

- Each product shows a maximum of 4 reviews at once but allows you scroll with footer navigation.
- Each product detail includes the number of review, but when there is no review yet, it shows "No Review Yet".

## Leave Review

When viewing a beer, you have an option leave a review for that beer. When reviewing, you would provide a 1 to 5 rating for the beer as well as an optional comment section below.

- Click "Leave Review" button to show review forms, you can submit or close forms.
- Username and star rating are required on form.
- The page automatically shows your recently left review.
- You can go back to view all Beers & ratings.

## Add New Beer

While viewing highest rated or recently reviewed items, you can add new items for rating and review. When adding new beers, you would be prompted to fill out details about the beer and then to submit or close form.

- Click "Add New Beer" button to show review forms, you can submit or close forms.
- A brewery is selected from list of Breweries featured on the app.
- Breweries are chosen and added to database my admin.
- Does not accept Alcohol per Volume (ABV) entries greater than 99.
- After adding a beer, you shall be taken to that beer's page to leave review.

## Admin Interface

---

Only an administrator can add Breweries. An administrator can also add, modify or delete beers and beer reviews.

## Features

---

**Project Name:** bv\_exercise, **App Name:** rankings **Preloaded data:** beerlist.json

## Folder Locations

- Template files: bv\_exercise/rankings/rankings/templates
- Static css: bv\_exercise/rankings/rankings/css
- Static images: bv\_exercise/rankings/rankings/products\_img

## Views

- Beer Ranking - All Reviews: `class ReviewView(TemplateView)`
- Beer details: `class ProductView(TemplateView)`

## Models

- **Brewery** - name `class Brewery(models.Model)`
- **Beer** - name, brewery, style, abv, description `class Beer(models.Model)`
- **Review** - date, username, ranking, comment, beer `class Review(models.Model)`

## Templates

- Beer Ranking - All Reviews: review.html
- Beer details: product.html

## Forms

- Add New Beer: ProductForm
- Leave Review: ReviewForm

## URLs

Used these URL pattern to the views

```
urlpatterns = [
    url(r'^$', views.ReviewView.as_view(), name='review'),
    url(r'^view-product/(?P<beer_id>[0-9]+)/$', views.ProductView.as_view(),
name='view-product'),
]
```

## Extensibility

---

In the future, it would be possible to add beer avatars by including a file input and render the address of the image to HTML template. In the meantime, I used an generic avatar/default for all Beers. Also, as the database expands, I've included an inactive footer pagination as an HTML comment.

## Other Notes

---

Focus outline of clickable elements have been designed to aid with accessibility.