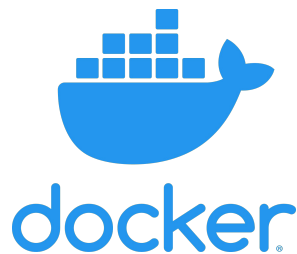


CS350 Milestone 2 - Code

Evan Duffield



Containerizing

- Team was having issues writing documentation that would apply to both Windows and Linux
- I containerized our Django application to streamline the process
- Will separate application from database, easy deployment, less commands to build and run.
 - Python virtual environment: 5 cmds to build and run
 - Docker: 2 cmds

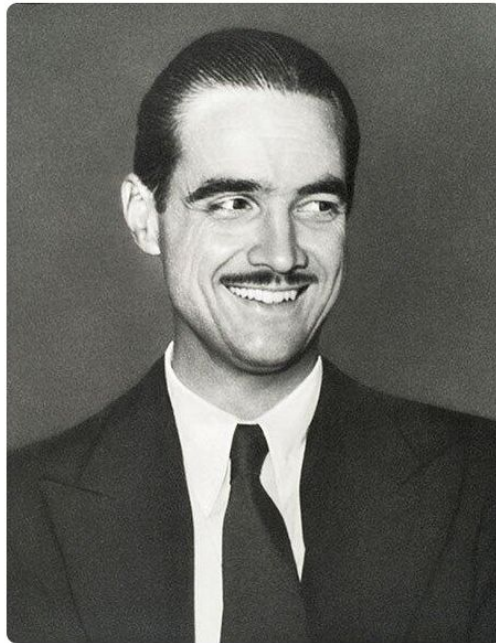
```
1  # Use an official Python runtime as a parent image
2  FROM python:3.9
3
4  # Set environment variables (adjust as needed)
5  ENV PYTHONDONTWRITEBYTECODE 1
6  ENV PYTHONUNBUFFERED 1
7
8  # Set the working directory in the container
9  WORKDIR /app
10
11 # Copy the requirements file into the container
12 COPY requirements.txt /app/
13
14 # Install project dependencies
15 RUN pip install -r requirements.txt
16
17 # Copy the rest of the application code into the container
18 COPY . /app/
19
20 # Expose the port the application will run on
21 EXPOSE 8000
22
23 # Start the Django application
```

The Iconic Squad

Meet This Guy!

Prototyping

- Half of group is in BACS350, where we created profile views in Django with data.
- Not fluent enough in frontend to implement adding and linking profiles, however. BACS350 project is a good milestone at this phase.



The stats:

Name: Howard Hughes

Strengths: Absolutely decimating a steak, staying in a movie theatre for 4 months

Weaknesses: OCD, Planes

Get This Guy @: </static/images/3.jpg>

Prototyping

- I decided to begin to implement the Python classes that will be used in our application, focusing on creating a User and Friends List class.
- Users are given unique hash to track account in database, Friends List object features sending and replying to friend requests.
- Basic testing script to see if methods set and return variables properly

```
27         return self.friends
28
29     def send_friend_request(self, userID):
30         if userID not in self.friends:
31             pass
32
33     def recieve_accept(self, userID):
34         self.friends.append(userID)
35
36     def recieve_friend_request(self, userID):
37         """ Prompt user
38         yes = True
39         if yes:
40             self.friends.append(userID)
41         else:
42             """ Send rejection
43             pass
44
45     def remove_friend(self, userID):
46         if userID in self.friends:
47             self.friends.remove(userID)
48
49
```

```
56 class User:
57     def __init__(self, username: str, password: str, email: str):
58         self.username = username
59         self.password = password
60         self.email = email
61
62         # ID will be generated by hashing the email, password,
63         self.epoch_registration = int(time.time())
64         self.id = self.make_hash()
65
66         # Attributes to be set later
67         self.firstname = None
68         self.lastname = None
69         self.title = None
70         self.pronouns = None
71         self.description = None
72
73
74         self.jobs = [] # List of Job Objects
75
76
77         self.friends = [] # New Friendslist object
78
79         """Hashes a unique id for the user"""
80     def make_hash(self):
```

Markdown Updates

- Updated Code directory in Milestone-2
- Filled out markdown forms alongside adding User.py and Testing.py

```
5
6  ## What I did for Milestone 2
7
8  ### Containerizing our application
9
10 Our team members were using a combination of either Windows
    or Linux machines. This became an issue during documentation
    as setting up the same environment on different OS required
    a lot of time to create directions. Using a containerization
    platform will allow us all to have the same development
    environment regardless of OS.
11
12 To do this, I decided to containerize our application using
    Docker. The setup can be seen in the Dockerfile in our
    project directory, and the commands to run the server are
    in the project readme. The benefit of this is that the team
    no longer need to configure a virtual environment for local
    development, and we can divide the application and database
    easily to help provide redundancy.
13
14 ### Prototype
15
16 Several of our team members are also in the BACS350 class,
    which recently completed a project to create profile views
    using templates and userdata. Creating and linking users
    together using a frontend is out of our scope at the moment
```

Thanks for watching!