Name	Maksims Kompanijecs
Student number	15306971
Programme	EC3
Module Code:	CA377
Assignment Title:	Programming Fundamentals (project)
Submission date:	16/12/18
Module Coordinator:	Jennifer Foster & Marija Berzbradica

I declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found at:

http://www.dcu.ie/info/regulations/plagiarism.shtml

https://www4.dcu.ie/students/az/plagiarism and/

or recommended in the assignment guidelines.

Name: Maksims Kompanijecs **Date:** 16/12/2018

Introduction

Link to my website: www.maksimk2.pythonanywhere.com/

Main idea of this project was to develop a web application called EAT@DCU. This application would allow users to search for places to eat across Dublin City University's campuses. This web app was built using Hypertext Markup Language, Cascading Style Sheets and Django framework. This wasn't my first time developing web applications using Django. Last year in CA229 module we developed an interactive website using Django framework. However last year I didn't use this language to the full potential and only used this framework to host the website locally.

This project was built and assessed in stages. We had 7 assignments to complete throughout the 12-week semester. We had around 2 weeks to complete each assignment, this spread out the workload nicely. This allowed to me to complete all the assignments and achieve full mark in every single task. It was very useful to know that you would receive instant feedback once you have pushed your code to the GitLab. You could either push your work to GitLab or run the tests internally to see whether all the tests have passed.

At times this project was a real challenge. At the start I could grasp my head around the whole concept of this project and how to go about doing it. There weren't many tutorials available online that could directly answer my questions, other than official Django documentation. This made it much more of a challenge and made me do a lot of reading and learning about Django Framework.

System Architecture

Django framework is seen Model Template View (MTV) Architecture.

- Model – this is our application data
- Template - how data is displayed
- View – what data is displayed

First, we had import models we wish to use in our views.py file.

```
from .models import Restaurant,Campus
```

models.py

```
class Restaurant(models.Model):
    restaurant_id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=100)
    location = models.CharField(max_length=100)
    campus_id = models.ForeignKey(Campus,on_delete = models.CASCADE)
    opening_hours = models.TimeField()
    closing_hours = models.TimeField()
    capacity = models.IntegerField()

    def __str__(self):
        return self.name
```

We modified our restaurants() function in views.py, which would filter the data based on campus_id and Boolean is_restaurant expression. This function distinguishes restaurants and cafes.

```
def restaurants(request):
    template = loader.get_template('eatatdcu/restaurants.html')
    campus_name = request.GET.get('campus','').lower()
    try:
        campus = Campus.objects.get(name=campus_name)
        restaurants = Restaurant.objects.filter(campus_id=campus,is_restaurant=True)
        cafes = Restaurant.objects.filter(campus_id=campus,is_restaurant=False)
    except Restaurant.DoesNotExist:
        return HttpResponse(template.render({'error':'No such restaurant'},request))
    except Campus.DoesNotExist:
        return HttpResponse(template.render({'error':'No such campus'},request))
    return HttpResponse(template.render({'restaurants':restaurants,'cafes':cafes},request))
```

When we modify restaurants.html, we can decide on how can display our data.

```
{% if error %}
   {{ error }}
{% else %}
   <h3>Restaurants</h3>
   {% if restaurants %}
      {% for r in restaurants %}
         <li>{{ r.name}}, {{r.location}}
         {% if r.is_staff_only %} (staff only!) {% endif %}
     <a href="{% url 'eatatdcu:specials' restaurant=r.name %}">Specials</a>
         </li>
      {% endfor %}
   {% else %}
      No restaurants found
   {% endif %}
```

When we access our restaurants.html page we can see data displayed like this.

Restaurants

- main restaurant, between albert college and the henry grattan building Specials
- 1838, albert college (staff only!) Specials
- nubar restaurant, beside gym Specials

EAT at DCU system architecture

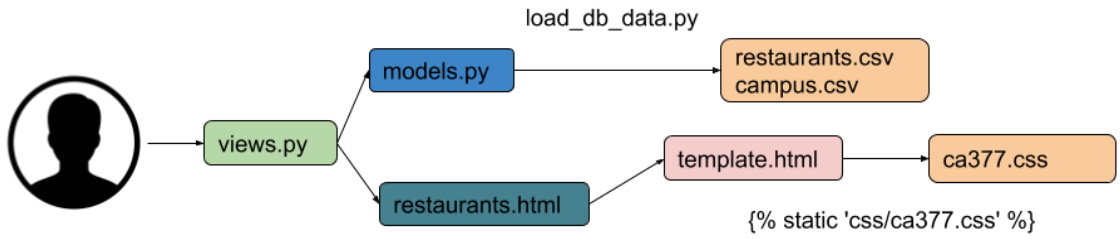


Fig.01 EAT at DCU system architecture

UI Description

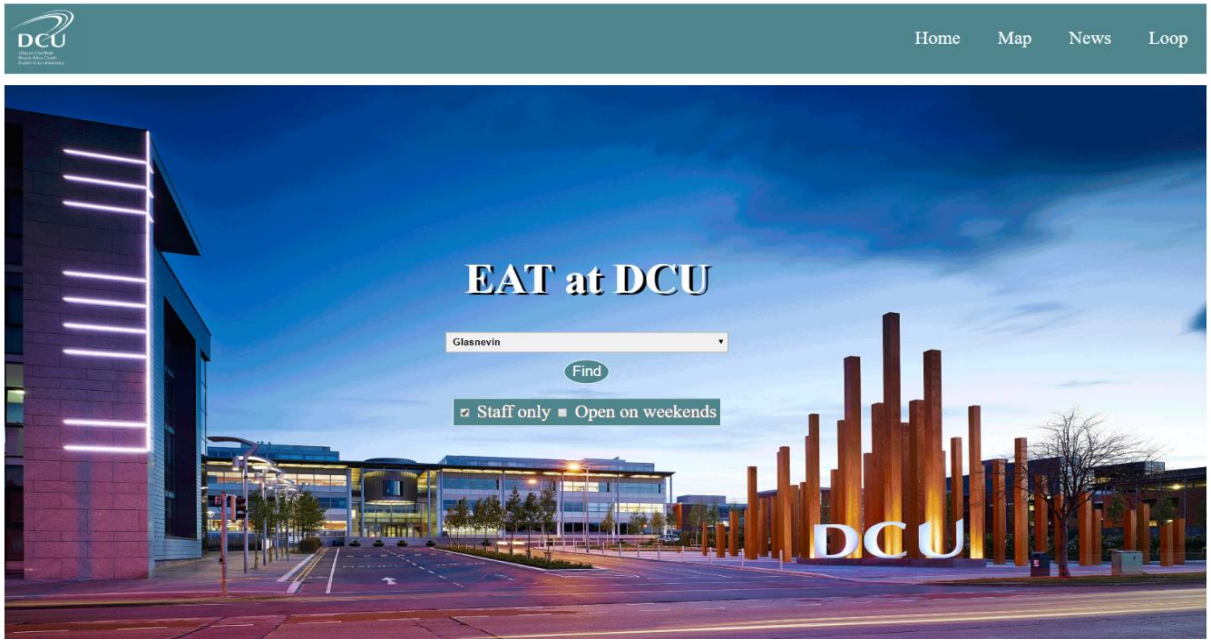


Fig.02 EAT at DCU homepage

As you can see above, this is my user interface. I followed branding guidelines and marking criteria for EAT@DCU, while developing this interface. I used the current DCU website as an inspiration to get colour palettes, fonts and logos. This ensured that my website fits in well with general DCU look and feel. I was provided with three user stories and I was given a task to create an appropriate interface.

I made it very easy for users that want checkout possible places to eat at DCU. The select boxes display the all possible types of input. This makes it very easy for all the users and ensures correct input. This would benefit users that are not so familiar with the campuses.

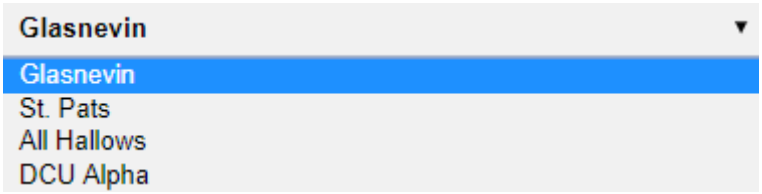


Fig.03 EAT at DCU dropdown

For the user stories I developed a staff only checkbox, that once checked would only display staff only places to eat at DCU. I also developed a checkbox that once checked would only display places that are open on the weekends. To see the daily specials on a given campus I added in a button that would display the daily special for a given restaurant. Based on the special of the day, the photo on that page would change accordingly. e.g. for pepperoni pizza daily special, a picture of pizza is displayed

Fig.04 EAT at DCU checkbox options

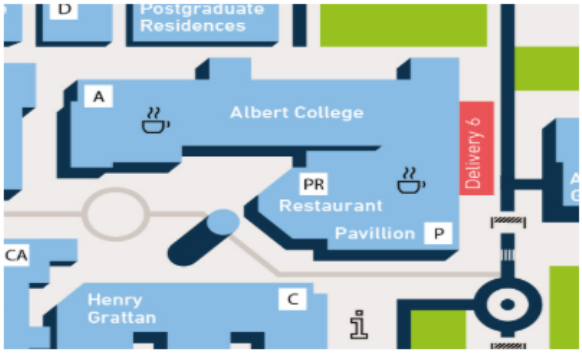
When searching for any places to eat on DCU campus. I added in a photo of the location where the restaurant or café is located. This benefits the users as they can clearly where the restaurant or café is

located. I also made some adjustments to the opening times in restaurants.html. Page now displays the opening times and closing times for every café or restaurant and also says if it is open on the weekends. This data is taken from restaurants.csv file and displayed in restaurants.html using Django templates. Average rating per restaurant was also added into the database and then displayed underneath the name.

main restaurant located in **between albert college and the henry grattan building**

3 rating out of 5

Click here to view todays Special

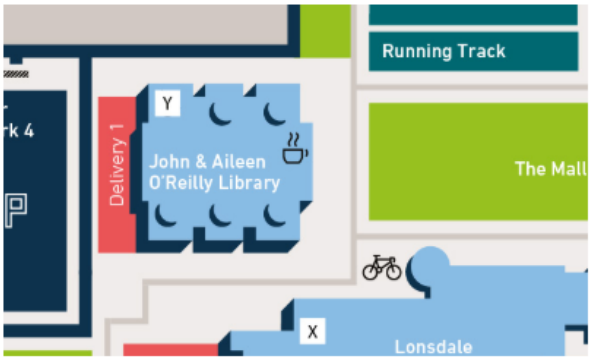


Opening hours: Monday: 7:30 a.m. - 5 p.m.
Tuesday: 7:30 a.m. - 5 p.m.
Wednesday: 7:30 a.m. - 5 p.m.
Thursday: 7:30 a.m. - 5 p.m.
Friday: 7:30 a.m. - 5 p.m.
Saturday: closed
Sunday: closed

Fig.05 Screenshot of main restaurant

library located in **library basement**

1 rating out of 5



Opening hours: Monday 9:30 a.m. - 8 p.m.
Tuesday 9:30 a.m. - 8 p.m.
Wednesday9:30 a.m. - 8 p.m.
Thursday 9:30 a.m. - 8 p.m.
Friday 9:30 a.m. - 8 p.m.
Saturday: 10 a.m. - 3 p.m.
Sunday: 10 a.m. - 3 p.m.

Fig.05 Screenshot of library café

Using GitLab

The link to my repository: www.gitlab.computing.dcu.ie/maksimk2/2019-ca377-EatAtDCU

For the entire lifecycle of this project we used GitLab as a source code management tool. Our code was hosted on DCU's GitLab server which is www.gitlab.computing.dcu.ie.

I made the full use of GitLab and I used it in many different ways. At the very start of this project we started off this project by forking a master repository. This allowed us to get any updates from the master repository from the original. We achieved this by fetching and merging from the master

branch. However, some conflicts had to be fixed manually. This was particularly the case when a new solution became available to us. We had to delete old code in the editor.

And of course, initially I had some problems using GitLab. Sometimes I would forget to pull the project and start working on it. This would cause issues when trying to push the project as there would be conflicts, as the branch would be up to date.

The biggest advantage that GitLab had to offer was that how easy it was access the code from different machines. I installed Git and Django at home, and there was no problem working on the code whether it was at home or in the labs.

Additional Functionality

The biggest feature regarding additional functionality that I implemented was an option to sort places to eat based on if they were open on weekends or were staff only. I took a very different approach to this. At first, I implemented the checkboxes and wasn't sure how I would make it work. I started to notice that every time the checkbox was checked the path of the URL would change. I researched how to make use of this and quickly figured out that you can base your output based off the URL. For example, the URL for Glasnevin restaurants that are staff only looks something like this:

/restaurants?campus=glasnevin&staff_only=on

So therefore when 'staff only' box would be checked. I would display staff only restaurants on Glasnevin campus. I achieved this by changing the contents of restaurants.html for any possible outcome. The code for Glasnevin staff only restaurants looks something like this.

```
{% if request.get_full_path == "/restaurants?campus=glasnevin&staff_only=on" %}
    {% for r in restaurants %}
        {% if r.is_staff_only %}

    DO SOMETHING

        {% endif %}
    {% endfor %}
{% endif %}
```

I also display a location of every restaurant or café beside the opening hours. This was done by making the use of {{r.name}} variable. In order to achieve this. I had to save different photos in my static folder which is located at static/img/maps corresponding with the name of the restaurants and cafes.

```
<img class="location" src="static/img/maps/{{r.name}}" alt="{{r.name}}">
```

For every instance of this for loop, {{r.name}} or changes as we keep going through the for loop. This allowed the images to change based on the name of the restaurant. The same idea worked for cafes, average ratings and daily specials.

Using JavaScript, I added in a button in the specials page. This button would allow users to redirect back to the previous page after seeing the daily specials for a specific restaurant. I figured out that this feature would be useful to the users that want to look at specials on particular campus. I achieved this by adding in a script which looks like this.

```
<div class="goback">
<button  onclick="goBack()">Go back to the restaurants page</button>
</div>
<script>
function goBack() {
    window.history.back();
}
</script>
```

Conclusion

Really enjoyed this module and I spent the most time working on this project. This project helped me to develop my programming skills. I now have much better understanding of different programming concepts, especially object orientated programming. My SQL skills have also improved and now I have much better understanding of Git and Unix shell scripting.

The tasks given were very well laid out. Most of the time we had around 2 weeks to complete a task, which gave me plenty of time to come up with a solution. The only task that I struggled with was A1: Database Part One. However, I still managed to come up with a solution that gave me 13 out of 15 marks. I will definitely use Django for any web application development projects in the future.

I am hoping to achieve a high mark in this module. It was a benefit to us how we got instant feedback on our code as soon as it was uploaded to GitLab. Without this I don't think I would have been able to do so well in this module. Sometimes when pushing my code to GitLab, the tests would fail. However, this gave me a chance to look back and fix my code.

Programming and development is definitely a career field I'd like to pursue once I finish my degree. But having secured an internship as a Business Analyst at Colgate-Palmolive that could possibly change my mind.

References

I mainly these types of resources to complete this project.

1. CA377 Programming Fundamentals Project wiki - https://gitlab.computing.dcu.ie/jfoster/2019-ca377-EatAtDCU/wikis/home
2. Stackoverflow - https://stackoverflow.com/
3. W3school Online Web Tutorials - https://www.w3schools.com/html/html_intro.asp
4. Django 1.11 official documentation - https://docs.djangoproject.com/en/1.11/
5. Youtube – www.youtube.com

How to filter queries in views.py and render them into the template
https://stackoverflow.com/questions/38863224/how-can-i-filter-queries-in-view-py-and-render-them-into-the-template

Django built-in template tags and filters
https://docs.djangoproject.com/en/1.11/ref/templates/builtins

Request paths in Django templates
https://stackoverflow.com/questions/15436407/request-path-in-django-template

Management of static files in Django
https://docs.djangoproject.com/en/1.11/howto/static-files/

How to position text over an image
https://www.w3schools.com/howto/howto_css_image_text.asp

History back() method
https://www.w3schools.com/jsref/met_his_back.asp

CSS Navigation Bar
https://www.w3schools.com/css/css_navbar.asp

Activating Models
https://www.youtube.com/watch?v=dONYOtb2ySI&list=PL6gx4Cwl9DGBlmzzFcLgDhKTTfNLfX1IK&index=8