**Artefacts List by Oliver Johnson**

Release 1

Login

I implemented the login and logout features for the website which involved creating a html template as well as functions in both views.py and lookup.py. This was a key feature for the website, as university students needed to be able to log on to their profiles, where parking permits, fines and other features were going to be bound to. The HTML is a simple setup which includes python if statements and utilizes the code within views.py which checks that the input email and password match those within the database. This is achieved by the method lookup.py.

Update to database

The updates I inputted into the database were to populate the site with users that had parking permits, violations as well as other sorts of violations. This helped make the site feel a bit more real than just having a test user and admin user. It also was more utilized when setting up global notifications and violations that appeared on the admin page, displaying how the page would work. In the release 1, database updates folder I included a cut version of the db.py that has the updates I inserted.

Update permit

The update permit was a featured added onto user profiles pages, this allowed the user to update their permit. Although it was intended for the user to update a lot of the displayed information, it was concluded to limit what they could change. So, the final version of update permit allows the user to update their vehicle type as well as whether their permit is monthly or yearly. The code for this is an update permit html template as well as relevant code within views.py and lookup.py

Update visitor page

The visitor page is a requirement of the website, to give those we are visitors to the college the ability to obtain a parking permit or report any violations. Fellow group member Kaelan, worked on setting up the initial visitor page html template, however asked me to complete it, including various email and name boxes, as well as a message box with a submit button. This was all done within the visitor html template.

Server aws setup

With the help and combined effort on both parts, both group members Sebastian and myself helped setup the website on amazon website services, to allow for the site to be pulled up on the internet anywhere, to give a better feel of having a proper website. In the folder is a screen shot of aws management of the server.

**Release 2**

Add rule

The add rule component was an addition to the rules page deemed needed, to allow admins greater flexibility in modifying the rules, giving them a necessary tool to easily add a new rule to the page without having to mess with any html. Add rule was achieved with adding a add button line the rules page html template writing up by Kaelan which directs the admin to an add rules html template, also accompanied with the relevant methods in both views.py and lookup.py

Update rule

The update rule feature, is the core tool that allows admins interact with the rules and regulations inside the browser. The rule allows admin to easily update any rule in the site with a click of a button, it takes the admin to a new page, which is a new update rules html template, that presents the attached title and description on the rule clicked, allowing the user to freely edit either, and once submitted will update the site.

Search functionality

This was a core feature, as an admin of the site, there could be tens of various health, smoking, overdue payments or other violations that require their attention, with managing so many requests it is key that they can search and filter for specific items they believe to be most important. The page itself is split into three sections, health, parking and other, and within these by using the search feature, the admin can further filter and better manage all violations they are dealing with.

Email notifications   
as part of a way to inform users of certain events, such as rules changes, overdue payments as well as payment confirmation, email notifications was desired. In certain functions a piece of code that creates and sends the email is placed, causing selected methods, when triggered to send out an email. A key part to the email setup however is defining the email and login details at the start of the code, within the artefact is a copy of the code used to define as well as a sample of how the email is implemented into another function.

Visitor page submit form

This was one of my final tasks and it was to complete the visitor page and have it so when it is submitted, it informs the user that an email has been sent to confirm and then an email is sent. The use of this allows for proper management of visitors, confirming through email that they have request a permit/as well as giving them the ability to report certain issues or message the admins even as a visitor. Although not as important as other parts, having the visitor page complete is an essential piece in the overall website to allow it to flow and work professionally.