

# ITRI615 - Computer Security

## Project Documentation

Affaan Muhammad - 33016763

Joshua Esterhuizen - 30285976

# Contents

<b>1</b>	<b>Installation and setup</b>	<b>2</b>
1.1	Project files . . . . .	2
1.2	Virtual Environment . . . . .	2
1.2.1	Creating a virtual environment . . . . .	3
1.2.2	Listing virtual environments . . . . .	3
1.2.3	Deleting a virtual environment . . . . .	3
1.2.4	Activating and deactivating virtual environments . . . . .	3
1.2.5	Listing Packages installed . . . . .	4
1.2.6	Using pip . . . . .	4
1.3	Frameworks and other packages . . . . .	5
1.3.1	Django . . . . .	5
1.3.2	Bootstrap . . . . .	5
1.3.3	Miscellaneous . . . . .	5
<b>2</b>	<b>Programming of artefact</b>	<b>6</b>
2.1	Development Tools . . . . .	6
2.1.1	Operating Systems . . . . .	6
2.1.2	IDEs . . . . .	6
2.1.3	Database Management Tools . . . . .	6
2.1.4	Hosting . . . . .	6
2.2	Prerequisites . . . . .	7
2.2.1	Project and Package Initialisation . . . . .	7
2.2.2	Settings and Admin . . . . .	7
<b>3</b>	<b>User manual</b>	<b>8</b>
<b>4</b>	<b>Reflection</b>	<b>9</b>
<b>5</b>	<b>Sources</b>	<b>10</b>

# Section 1

## Installation and setup

### 1.1 Project files

The project files can be found on the following GitHub link:

<https://github.com/AM-ops/SecurityProject>

This was our main code repository. We both have been updating the code as we went along and added details and bug fixes to the project.

To copy the code to your own machine, follow the following steps:

1. Make sure Git is installed. If not it can be downloaded from here:  
<https://git-scm.com/>
2. Create an empty directory where the code can be copied to
3. Run the following command:

```
git clone https://github.com/AM-ops/SecurityProject.git
```

### 1.2 Virtual Environment

There are multiple advantages of using virtual environments when creating software. The primary reason being we create a layer of separation and abstraction between our host machine's files and our software project.

We made use of a Python virtual environment which was handled by Anaconda. This can be downloaded from the following link:

<https://www.anaconda.com/products/individual>

### 1.2.1 Creating a virtual environment

Once Anaconda was installed the following commands were run in the terminal to create a virtual environment called myDjangoEnv.

```
conda create --name myDjangoEnv
```

Depending on the version of Anaconda installed you might have to use a leading underscore on Windows machines. The same will apply for commands further down. Below is a demonstration.

```
_conda create --name myDjangoEnv
```

### 1.2.2 Listing virtual environments

To list all virtual environments on your host machine run the following command.

```
conda info --envs
```

or

```
conda env list
```

### 1.2.3 Deleting a virtual environment

To delete a virtual environment run the following commands.

```
conda remove --name <name_of_virtual_environment> --all
```

or

```
conda env remove --name <name_of_virtual_environment>
```

### 1.2.4 Activating and deactivating virtual environments

To activate an environment run the following commands for Windows.

```
conda activate <name_of_virtual_environment>
```

For Linux and MacOS the command is as follows.

```
source activate <name_of_virtual_environment>
```

Once the environment is activated your terminal should change. By default, the active environment, is shown in parentheses () or brackets [] at the beginning of your command prompt as shown below.

```
(<name_of_virtual_environment>) >_
```

Depending on your version of Anaconda to deactivate your environment the commands for Windows is.

```
deactivate
```

or

```
conda deactivate
```

For Linux and MacOS the command will be

```
source deactivate
```

## 1.2.5 Listing Packages installed

To list all the packages you have installed in an environment there are two methods of listing them. First, if the environment is not activated run the following.

```
conda list -n <name_of_virtual_environment>
```

Secondly, if the environment is activated, then simply run the following.

```
conda list
```

## 1.2.6 Using pip

Due to the fact that Python is being used for the project it is always necessary to make sure pip is installed and functioning. If it is not then run the following commands.

```
conda install -n <name_of_virtual_environment> pip
```

## 1.3 Frameworks and other packages

### 1.3.1 Django

The primary framework used for development in this project was Django. This is a python based Web framework. The documentation for it can be found here:

<https://docs.djangoproject.com/en/3.2/>

### 1.3.2 Bootstrap

Bootstrap is Cascading Style Sheets (CSS) Framework which allows for simple, elegant, and responsive Graphical User Interfaces to be developed for the Web. The documentation for it can be found here:

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

For a more seamless integration of Bootstrap with the Django Framework an additional package called `django-crispy-forms` was also installed. Its documentation can be found here:

<https://django-crispy-forms.readthedocs.io/en/latest/>

### 1.3.3 Miscellaneous

For typesetting of this documentation, L<sup>A</sup>T<sub>E</sub>X was utilised. Additionally, a L<sup>A</sup>T<sub>E</sub>X package called `minted` was used to typeset code in this documentation. Its homepage is located at:

<https://www.ctan.org/pkg/minted>

Lastly, to typeset code within the HTML pages of our project the JavaScript library called `Rainbow` was implemented. The GitHub link for that is located at:

<https://github.com/ccampbell/rainbow>

## Section 2

# Programming of artefact

## 2.1 Development Tools

### 2.1.1 Operating Systems

The primary systems on which development was done was Linux and Windows 10. The same systems were utilised for testing and bug fixing purposes.

### 2.1.2 IDEs

For the purpose of coding the following two Integrated Development Environments were used:

1. Atom. It can be downloaded from: <https://atom.io/>
2. Visual Studio Code, also known as VSCode. It can be downloaded from here: <https://code.visualstudio.com/>

### 2.1.3 Database Management Tools

For the purposes of database management, TablePlus was the main software we utilised. It was used to see if our Django models and cryptographic schemes were correctly implemented. TablePlus can be downloaded from: <https://tableplus.com/>

### 2.1.4 Hosting

Due to a number of constraints we landed up running our project locally. The server was `localhost` and the port number was 8000. Therefore the link where we ran our project was: <http://127.0.0.1:8000>

## **2.2 Prerequisites**

### **2.2.1 Project and Package Initialisation**

### **2.2.2 Settings and Admin**



## Section 3

### User manual

## Section 4

## Reflection

## Section 5

### Sources

<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

<https://www.udemy.com/course/python-and-django-full-stack-web-developer-bootcamp/>

<https://docs.djangoproject.com/en/3.2/>