**Software Implementation and Testing Document**

**For**

**Group <13>**

Version 2.0

**Authors**:

Taj Ali

Harege Abay

Tobechukwu Osakwe

Vinson Thomas

# Programming Languages

*Programming Languages:*

* *Python: Used as the interpreter and base language for the project and was mutually agreed upon by each group member*
* *Html: Used in the creation and editing of web pages in the group’s website*
* *CSS: Used in editing style of Html web pages*
* JavaScript: *Used to add additional client-side functionality*

# Platforms, APIs, Databases, and other technologies used

*Platforms:*

* *Pycharm IDE: Used for the implementation of the Django project*
* *Sqlite: used for relational tables to store data*
* *Django: This was used and needed for the web development portion of our group’s project*
* *Django Admin: used to manage, create, and edit database rows, instead of having to manually edit them using SQL*

# Execution-based Functional Testing

We did not use any sort of automated unit testing. Instead, we tested functionality by writing code, and then testing it by hand in the Web browser.

# Execution-based Non-Functional Testing

We tested nonfunctional requirements, such as security, mostly in the same way we tested our functional requirements. As we have no automated testing infrastructure, we tested things like logging in by performing actions manually.

# Non-Execution-based Testing

We did not do any non-execution-based testing.