

---

# Technology Requirements Documentation

Technology Requirements for  
Domain Pulse

---

Ctrl Alt Defeat

2023/05/23

# Contents

<b>1</b>	<b>Technology Requirements</b>	<b>2</b>
1.1	Development Environment . . . . .	2
1.2	Version Control . . . . .	2
1.3	Programming Languages . . . . .	2
1.4	Frameworks and Libraries . . . . .	2
1.5	Database Systems . . . . .	2
1.6	Testing and Quality Assurance Tools . . . . .	2
1.7	Deployment and Infrastructure . . . . .	3
1.8	Collaboration and Communication Tools . . . . .	3
1.9	Security and Encryption . . . . .	3
1.10	Continuous Integration and Deployment Tools . . . . .	3

# **1 Technology Requirements**

## **1.1 Development Environment**

For development, our team members shall be developing within VS Code on Linux to ensure consistency within the produced code and testing of our software. Having all members using VS Code also allows for the use of software such as Live Share for collaborative development to improve efficiency within development.

## **1.2 Version Control**

Git will be used for version control, using a clear and simple branching strategy that allows members to easily work on distinct components of our system while also allowing the reverting to previous versions as a 'fail-safe'. Meanwhile, GitHub will be used to host the Git repository.

## **1.3 Programming Languages**

Within our team, many (if not all) members have a high proficiency in coding in Python. Python is considered one of the best programming languages for Machine Learning and data analysis, which are aspects on which our system will heavily rely. Therefore, Python is our programming language of choice.

## **1.4 Frameworks and Libraries**

We have decided to use Django as our web framework for the creation of our app due to its simplicity, efficiency, and secure data transmission capabilities. Within our Django project, we will be utilizing Python's proficiency in data analysis and machine learning by using the powerful Vader and Grafana libraries, which allow us to perform sentiment analysis on text and visualize our data, respectively. For the front-end development, we will be using Angular due to its versatility and ability to create high-quality user interfaces.

## **1.5 Database Systems**

For our database systems, we will be using MongoDB as a document-based NoSQL database for data collection of comments, posts, and other content related to user data. Additionally, PostgreSQL will be used as an SQL database for storing user profiles and authentication data for ease of access and querying.

## **1.6 Testing and Quality Assurance Tools**

We will perform various forms of testing to ensure the quality of our software. For front-end testing, Cypress will be used, and for unit testing, we will utilize the built-in unittest library of Python, which is recommended for Django.

## **1.7 Deployment and Infrastructure**

For deployment, our clients have provided a Linux-based virtual machine where we can deploy our software. We will use one virtual machine for the testing environment and another for the production environment to ensure a separation of concerns. Additionally, a domain may be acquired for ease of customer access to the software.

## **1.8 Collaboration and Communication Tools**

To ensure effective communication within our team, we have a private WhatsApp group for important announcements, a private Discord server for more specific announcements and discussions, and a GitHub Project Board to track work progress. For communication with our clients, we have set up a Slack workspace for quick communication.

## **1.9 Security and Encryption**

Our system is secured using technologies such as Django's authentication system, which encrypts user data. We prioritize the use of POST over GET for more secure data transmission. Our virtual machine is protected by extensive firewalls to prevent foreign attacks, and access is restricted to authorized members of our team using SSH and private login details.

## **1.10 Continuous Integration and Deployment Tools**

We will utilize CI/CD technologies such as Codecov and GitHub Actions to ensure thorough testing and checking before deployment. GitHub Actions will automate the building, testing, and deployment processes, while Codecov will provide insights into test results and failures. We have been provided with separate production and testing environments to deploy and check the application's functionality.