

# Argupedia - Appendix

6CCS3PRJ – Individual Project

Bsc Computer Science

April 2020

---

---

**Author: A K M Naharul Hayat**

**Student Number: 1750583**

**K Number: K1764014**

---

---

**Supervisor: Dr. Sanjay Modgil**

## Contents

<b><u>A.1 EXTERNAL LIBRARIES USED .....</u></b>	<b><u>3</u></b>
<b><u>A.2 PROJECT RUNNING INSTRUCTIONS .....</u></b>	<b><u>5</u></b>
<b><u>A.3 TESTING SCREENSHOTS.....</u></b>	<b><u>6</u></b>
<b><u>A.4 CODE LISTING .....</u></b>	<b><u>23</u></b>

## A.1 External Libraries used

External libraries used in the project (except the Django framework itself) is located in the path: `'Lib\site-packages'`. The external libraries & framework used in the project along with their purpose is listed below:

### Django - 2.2.7

The framework for the project which lays out the foundational structure of the architecture of the application and provides essential security libraries for web application development.

Source: <https://www.djangoproject.com>

### Django-Mptt - 0.9.1

This library provides a tree data structure which perfectly integrates with Django model. It is used to represent debates in a graph structure in the database

Source: <https://django-mptt.readthedocs.io/en/latest/>

### Bootstrap – 4.3.1

Used to provide a consistent and coherent front-end styling for the application.

Source: <https://getbootstrap.com>

### jQuery – 3.3.1

Allows sections in pages of the application to update without the need to reload the entire page. In addition, it is used to make asynchronous requests for upvoting/downvoting an argument.

Source: <https://jquery.com>

### Bleach – 3.0.2

This library allows sanitization of HTML content. It allows filtering HTML tags, attributes and styles to a permissible subset defined. It is used when formatting user input of an argument to highlight headers of scheme structure.

Source: <https://pypi.org/project/django-bleach/>

### D3.JS – 4.0

This is a JavaScript based library used to visualize debates in the form of directed graph.

Source: <https://d3js.org>

### Django-JS-Asset – 1.2.2

Allows insertion of additional attributes when importing scripts in HTML. Is used when importing 'D3'.

Source: <https://github.com/matthiask/django-js-asset>

### Markdown – 3.0.1

Allows conversion of 'Markdown' formatted text to HTML. It is used when formatting user input of an argument to highlight headers of scheme structure.

Source: <https://pypi.org/project/django-markdown/>

### Pytz – 2019.3

Allows accurate and cross platform time zone calculations. It is used to timestamp arguments in debates.

Source: <https://pypi.org/project/pytz/>

Six – 1.13.0

Allows bridging the gap between python version 2 and 3. It offers utility functions to smooth the gaps between the Python versions with the purpose of writing Python code consistent with all Python versions.

Source: <https://pypi.org/project/django-six/>

SQLParse – 0.3.0

It offers parsing and formatting support for SQL statements.

Source: <https://pypi.org/project/sqlparse/>

WebEncodings – 0.5.1

Allows compatibility with legacy web-content in charset interpretation.

Source: <https://pypi.org/project/webencodings/>

Other External materials used

The tree style commenting system was heavily inspired from the project at the following source:

<https://github.com/MakuZo/bloggy>

## A.2 Project Running Instructions

- Extract the project.
- Navigate to the path of the project using command prompt:
  - `cd <project location>`
- Navigate to the scripts folder using command prompt:
  - `cd Scripts`
- Activate virtual environment:
  - `activate.bat`
- Back out of the folder into the root folder:
  - `cd ..`
- Run the following command:
  - `python manage.py runserver`
- This will start the web-server. A URL will be displayed in the command prompt. Copy the URL into a browser to access the website.

## A.3 Testing Screenshots

### A.3.1

# Register

Username:

testuser

Password:

...

Password confirmation:

...

- This password is too short. It must contain at least 8 characters.
- This password is too common.
- This password is entirely numeric.

Register

Cancel

### A.3.2

---

## Register

Username:

Password:

Password confirmation:

- This password is too common.
- This password is entirely numeric.

### A.3.3

---

## Register

Username:

Password:

Password confirmation:

# Login

Username:

naharul

Password:

...

Login

Cancel

naharul ▼

Create a Debate Topic

## A.3.4

naharul

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

**algorithm:** pbkdf2\_sha256 **iterations:** 150000 **salt:** 81g9u1\*\*\*\*\* **hash:** omubJG\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).



A.3.5

## Should weed be legalized?

[Login](#) or [Register](#) to contribute to debate!

**naharul** Votes: 1

### **In Situation:**

Weed being illegal

### **Doing Action:**

Legalizing weed

### **Achieves Goal:**

sovereignty

### **Promotes Value:**

being high

A.3.6

## Select Argument Scheme

Action Scheme

Expert opinion

Popular Opinion

Position to Know

Argument from Analogy

Correlation to Cause

---

## Provide your argument

Topic Title:

test action scheme

In Situation:

test action scheme

Doing Action:

test action scheme

Achieves Goal:

test action scheme

Promotes Value:

test action scheme

Submit

Cancel

## test action scheme

[naharul](#) Votes: 1  - April 20, 2020, 6:33 p.m. [counter](#)

**In Situation:**

test action scheme

**Doing Action:**

test action scheme

**Achieves Goal:**

test action scheme

**Promotes Value:**

test action scheme

## Provide your argument

Topic Title:

test expert opinion scheme

Expert Source:

test expert opinion scheme

Expert Area:

test expert opinion scheme

Expert Assertion:

test expert opinion scheme

Submit

Cancel

## test expert opinion scheme

**naharul** Votes: 1  - April 20, 2020, 6:34 p.m. [counter](#)

**Expert Source:**

test expert opinion scheme

**Expert Area:**

test expert opinion scheme

**Expert Assertion:**

test expert opinion scheme

## Provide your argument

Topic Title:

popular opinion scheme test

Majority Group:

popular opinion scheme test

Fact Believed by the Majority:

popular opinion scheme test


Conclusion based on belief:

popular opinion scheme test

Submit

Cancel

## popular opinion scheme test

**naharul** Votes: 1  - April 20, 2020, 6:36 p.m. [counter](#)

**Majority Group:**

popular opinion scheme test

**Fact Believed by the Majority:**

popular opinion scheme test

**Conclusion based on belief:**

popular opinion scheme test

## Provide your argument

Topic Title:

position to know test

'Position to Know' Source:

position to know test

Domain of Source:

position to know test


Assertion made by source:

position to know test

Submit

Cancel

## position to know test

[naharul](#) Votes: 1  - April 20, 2020, 6:37 p.m. [counter](#)

### 'Position to Know' Source:

position to know test

### Domain of Source:

position to know test

### Assertion made by source:

position to know test

## Provide your argument

Topic Title:

Arg from analogy test

Two cases which are similar:

Arg from analogy test

Fact true in both cases:

Arg from analogy test


Conclusion based on correlation:

Arg from analogy test

Submit

Cancel

## Arg from analogy test

[naharul](#) Votes: 1  - April 20, 2020, 6:38 p.m. [counter](#)

### Two cases which are similar:

Arg from analogy test

### Fact true in both cases:

Arg from analogy test

### Conclusion based on correlation:

Arg from analogy test

## Provide your argument

Topic Title:

arg correlation to cause test

Historical data/experience:

arg correlation to cause test

Suggestion based on data:

arg correlation to cause test

Conclusion:

arg correlation to cause test

Submit

Cancel

## arg correlation to cause test

**naharul** Votes: 1  - April 20, 2020, 6:40 p.m. [counter](#)

**Historical data/experience:**

arg correlation to cause test

**Suggestion based on data:**

arg correlation to cause test

**Conclusion:**

arg correlation to cause test

### A.3.7

#### Original Argument

**Historical data/experience:**

arg correlation to cause test

**Suggestion based on data:**

arg correlation to cause test

**Conclusion:**

arg correlation to cause test

#### Counter Argument

Critique Position:

Select

Original Argument	Counter Argument
<p><b>Historical data/experience:</b> arg correlation to cause test</p> <p><b>Suggestion based on data:</b> arg correlation to cause test</p> <p><b>Conclusion:</b> arg correlation to cause test</p>	<p><b>Critique Position:</b></p> <div>Method used to establish correlation is questionable.</div> <p><b>Choose Counter-Argument Scheme</b></p> <div>Action Scheme</div> <div>Expert opinion</div> <div>Popular Opinion</div> <div>Position to Know</div> <div>Argument from Analogy</div> <div>Correlation to Cause</div>

Original Argument	Provide your counter-argument
<p><b>Historical data/experience:</b> arg correlation to cause test</p> <p><b>Suggestion based on data:</b> arg correlation to cause test</p> <p><b>Conclusion:</b> arg correlation to cause test</p>	<p><b>Critique Position:</b></p> <div>Method used to establish correlation is questionable.</div> <p><b>In Situation:</b></p> <div>test counter argument</div> <p><b>Doing Action:</b></p> <div>test counter argument</div> <p><b>Achieves Goal:</b></p> <div>test counter argument</div> <p><b>Promotes Value:</b></p> <div>test counter argument</div> <div><div>Submit</div><div>Cancel</div></div>

## arg correlation to cause test

**naharul** Votes: 1 + - April 20, 2020, 6:40 p.m. [counter](#)

### Historical data/experience:

arg correlation to cause test

### Suggestion based on data:

arg correlation to cause test

### Conclusion:

arg correlation to cause test

**test** Votes: 1 + - April 20, 2020, 6:49 p.m. [counter](#)

Critique Position: *Method used to establish correlation is questionable.*

### In Situation:

test counter argument

### Doing Action:

test counter argument

### Achieves Goal:

test counter argument

### Promotes Value:

test counter argument

### Original Argument

Critique Position: *Method used to establish correlation is questionable.*

#### In Situation:

test counter argument

#### Doing Action:

test counter argument

#### Achieves Goal:

test counter argument

#### Promotes Value:

test counter argument

### Counter Argument

#### Critique Position:

The action won't achieve the goals described.

#### Choose Counter-Argument Scheme

Action Scheme

Expert opinion

Popular Opinion

Position to Know

Argument from Analogy

Correlation to Cause



### Original Argument

Critique Position: *Method used to establish correlation is questionable.*

**In Situation:**

test counter argument

**Doing Action:**

test counter argument

**Achieves Goal:**

test counter argument

**Promotes Value:**

test counter argument

### Provide your counter-argument

**Critique Position:**

The action won't achieve the goals described.

**Majority Group:**

test counter-counter argument

**Fact Believed by the Majority:**

test counter-counter argument

**Conclusion based on belief:**

test counter-counter argument

Submit

Cancel

## arg correlation to cause test

**naharul** Votes: 1  - April 20, 2020, 6:40 p.m. [counter](#)

**Historical data/experience:**


arg correlation to cause test

**Suggestion based on data:**

arg correlation to cause test

**Conclusion:**

arg correlation to cause test

**test** Votes: 1  - April 20, 2020, 6:49 p.m. [counter](#)

Critique Position: *Method used to establish correlation is questionable.*

**In Situation:**

test counter argument

**Doing Action:**

test counter argument

**Achieves Goal:**

test counter argument

**Promotes Value:**

test counter argument

**naharul** Votes: 1  - April 20, 2020, 6:53 p.m. [counter](#)

Critique Position: *The action won't achieve the goals described.*

**Majority Group:**

test counter-counter argument

**Fact Believed by the Majority:**

test counter-counter argument

**Conclusion based on belief:**

test counter-counter argument

### A.3.8

## Provide your argument

Topic Title:

test incomplete

In Situation:

test incomplete

Doing Action:

test incomplete

Achieves Goal:

test incomplete

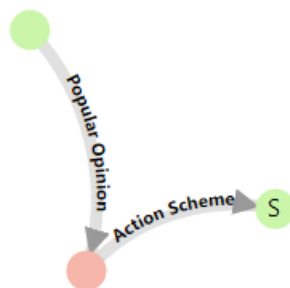
Promotes Value:

|

! Please fill out this field.

### A.3.9

■ In  
■ Out  
■ Undec



## Should weed be legalized?

**naharul** Votes: 1  - March 9, 2020, 9:13 p.m. [counter](#)

**In Situation:**

Weed being illegal

**Doing Action:**


Legalizing weed

**Achieves Goal:**

sovereignty

**Promotes Value:**

being high

**naharul** Votes: 1  - April 19, 2020, 10:01 p.m. [counter](#)

Critique Position: *Action will have undesirable consequence.*

**Two cases which are similar:**


ddddddddd

**Fact true in both cases:**

ddddddddd

**Conclusion based on correlation:**

ddddddddddd

**naharul** Votes: 1  - March 24, 2020, 4:50 p.m. [counter](#)

Critique Position: *The action won't have the goals described.*

**Expert Source:**


UCL some department

**Expert Area:**

drugs

**Expert Assertion:**

legailizadonasd

**naharul** Votes: 1  - March 24, 2020, 4:08 p.m. [counter](#)

Critique Position: *The goal doesn't realize the value.*

**Expert Source:**


kaksjdkasd

**Expert Area:**

asdasdasd

**Expert Assertion:**

asddfasdf

naharul Votes: 1  - March 10, 2020, 12:45 a.m. [counter](#)

Critique Position: *The situation described is not true.*

**Expert Source:**


sads

**Expert Area:**

asdasd

**Expert Assertion:**

asdasdasd

naharul Votes: -1  - March 9, 2020, 11:10 p.m. [counter](#)

Critique Position: *The situation described is not true.*

**Expert Source:**


a

**Expert Area:**

a

**Expert Assertion:**

a

naharul Votes: 0  - March 9, 2020, 11:10 p.m. [counter](#)

Critique Position: *Expert assertion is incorrect.*

**Expert Source:**


as

**Expert Area:**

as

**Expert Assertion:**

as

naharul Votes: 1  - March 10, 2020, 12:27 a.m. [counter](#)

Critique Position: *Expert source is not credible.*

**Expert Source:**

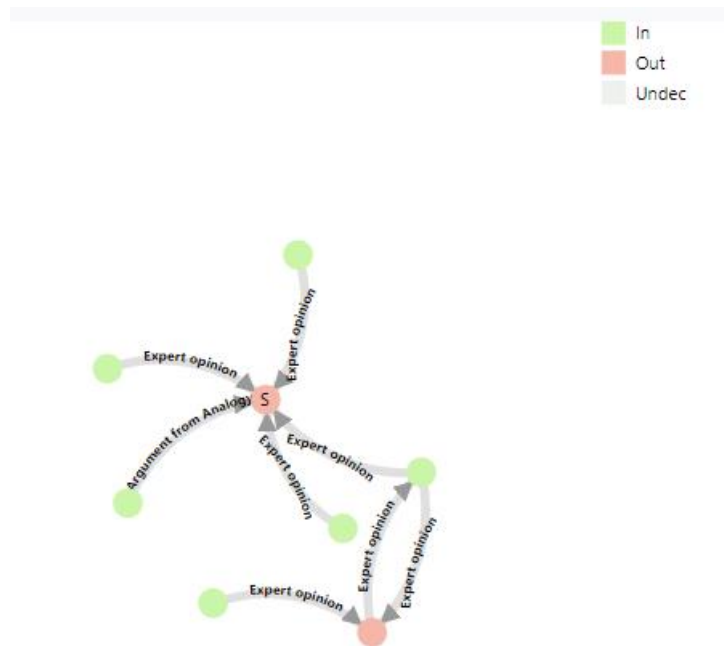
asd

**Expert Area:**

asda

**Expert Assertion:**

asdas



A.3.11

## Should weed be legalized?

**naharul** Votes: **1** + – March 9, 2020, 9:13 p.m. [counter](#)

### **In Situation:**

Weed being illegal

### **Doing Action:**

Legalizing weed

### **Achieves Goal:**

sovereignty

### **Promotes Value:**

being high

## Should weed be legalized?

**naharul** Votes: 0  - March 9, 2020, 9:13 p.m. [counter](#)

### **In Situation:**

Weed being illegal

### **Doing Action:**

Legalizing weed

### **Achieves Goal:**

sovereignty

### **Promotes Value:**

being high

## Should weed be legalized?

**naharul** Votes: -1  + March 9, 2020, 9:13 p.m. [counter](#)

### **In Situation:**

Weed being illegal

### **Doing Action:**

Legalizing weed

### **Achieves Goal:**

sovereignty

### **Promotes Value:**

being high

## A.4 Code Listing

\templates\base.html

```
{% load humanize %}
{% load static %}
<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
U02eT0CpHqD5JQ6hJty5KVphtPhzWj9WO1c1HTMGA3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
    <link rel="stylesheet" href="{% static '/argupedia/styles.css' %}" />
    <title>{% block title %}{% endblock %}</title>
</head>
<body>

<div id="app">
    {% include 'components/navbar.html' %}
    <div class="container-fluid">
        {% include 'components/messages.html' %}
        {% block content %}

        {% endblock %}
    </div>
</div>

</body>
{% include 'components/footer.html' %}
</html>
```

\templates\signup.html

```
{% extends 'base.html' %}

{% block title %}Sign Up{% endblock %}

{% block content %}

    <div class="container center_div">
        <div class="col-md-8 centered" style="padding-left : 300px; padding-bottom
: 40px">
            <div class="form-area">
                <form method="post">
                    {% csrf_token %}
                    <div class="container center_div pl-3">
                        <h3 style="margin-bottom: 15px; text-align:
center;">Register</h3>
                        {% for field in form %}
                            <div class="form-group">
                                <label for="text">{{ field.label_tag }}</label>

                                {% if 'password' in field.id_for_label %}
                                    <input type="password" class="form-control"
id="{{ field.id_for_label }}" name="{{ field.html_name }}" value="">
                                {% else %}
                                    <input type="text" class="form-control" id="{{
field.id_for_label }}" name="{{ field.html_name }}" value="">
                                {% endif %}

                            </div>
                            {% if field.errors %}
                                <div class="alert alert-danger" id="alert">
                                    {{ field.errors }}
                                </div>
                            {% endif %}
                        {% endfor %}
                        <input type="submit" name="submit" class="btn btn-outline-
dark" style="width:120px; padding-left:10px;padding-bottom:5px;" role="button"
value="Register"/>
                        <a href="{% url 'home' %}" class="btn btn-outline-dark"
style="width:120px; padding-left:10px;padding-bottom:5px;">Cancel</a>
                    </div>
                </form>
            </div>
        </div>
    </div>

{% endblock %}
```



```

\templates\registration\login.html

{% extends 'base.html' %}

{% block title %}Login{% endblock %}

{% block content %}
    <div class="container center_div">
        <div class="col-md-8 centered" style="padding-left : 300px; padding-bottom
: 40px">
            <div class="form-area">
                <form method="post">
                    {% csrf_token %}
                    <div class="container center_div pl-3">
                        <h3 style="margin-bottom: 15px; text-align:
center;">Login</h3>
                        {% for field in form %}
                            <div class="form-group">
                                <label for="text">{{ field.label_tag }}</label>

                                {% if 'password' in field.id_for_label %}
                                    <input type="password" class="form-control" id="{{
field.id_for_label }}" name="{{ field.html_name }}" value="">
                                {% else %}
                                    <input type="text" class="form-control" id="{{
field.id_for_label }}" name="{{ field.html_name }}" value="{{ field.errors }}">
                                {% endif %}

                                {% if field.help_text %}
                                    <p class="help">{{ field.help_text|safe }}</p>
                                {% endif %}

                            </div>
                            {% if field.errors %}
                                <div class="alert alert-success" id="danger">
                                    {{ field.errors }}
                                </div>
                            {% endif %}
                        {% endfor %}
                        <input type="submit" name="submit" class="btn btn-outline-
dark" style="width:120px; padding-left:10px;padding-bottom:5px;" role="button"
value="Login"/>
                        <a href="{% url 'home' %}" class="btn btn-outline-dark"
style="width:120px; padding-left:10px;padding-bottom:5px;">Cancel</a>
                    </form>
                </div>
            </div>
        </div>
    {% endblock %}

```

\templates\components\footer.html

```
<footer class="section footer-classic context-dark bg-image" style="padding-  
bottom:20px;">  
  <hr>  
  <div class="container">  
    <div class="row justify-content-center">  
      <span>© </span><span class="copyright-  
year">2020</span><span> </span><span>Argupedia - Individual Project</span>.  
    </div>  
    <div class="row justify-content-center">  
      </span><span>A K M Naharul Hayat, 1750583, K1764014</span>  
    </div>  
  </div>  
</footer>
```

\templates\components\messages.html

```
{% if errors.count > 0 %}
    {% for obj in errors %}
        <div class="alert alert-danger" id="alert">
            {{obj}}
        </div>
    {% endfor %}
{% endif %}

{% if success.count > 0 %}
    {% for obj in success %}
        <div class="alert alert-success" id="alert">
            {{ obj }}
        </div>
    {% endfor %}
{% endif %}
```

\templates\components\navbar.html

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo03" aria-controls="navbarTogglerDemo03" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="{% url 'home' %}">Argupedia</a>

  <div class="collapse navbar-collapse" id="navbarTogglerDemo03">
    <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
      <li class="nav-item">
        <a class="nav-link" href="{% url 'home' %}">Home <span class="sr-
only">(current)</span></a>
      </li>
      {% if user.is_authenticated %}
      <li class="nav-item">
        <a class="nav-link" href="{% url 'my-discussions-view'
user.username %}">Topics Engaged In</a>
      </li>
      {% endif %}
    </ul>
    {% if user.is_authenticated %}
    <ul class="inline my-2 my-lg-0" style="list-style-type: none;">
      <li class="nav-item dropdown ml-auto">
        <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">{{ user.username }}</a>
        <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
          <a class="dropdown-item" href="{% url 'logout' %}">Log out</a>
        </div>
      </li>
    </ul>
    {% else %}
    <form class="form-inline my-2 my-lg-0">
      <a href="{% url 'signup' %}" class="btn btn-light my-2 my-sm-0 mr-
1">Register</a>
      <a href="{% url 'login' %}" class="btn btn-light my-2 my-sm-
0">Login</a>
      <a href="/admin" class="btn btn-light my-2 my-sm-0">Admin
Login</a>
    </form>
    {% endif %}
  </div>
</nav>
```

```

\project\settings.py

"""
Django settings for project project.

Generated by 'django-admin startproject' using Django 2.2.7.

For more information on this file, see
https://docs.djangoproject.com/en/2.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.2/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'y*4=pp#oa57=%5*mp#v1jbgqp1qy3zmf&0lfid%arz9m+w7(ea'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'accounts.apps.AccountsConfig',
    'django.contrib.humanize',
    'mptt',
    'django.contrib.sites',
    'argupedia',
]
SITE_ID = 1
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'project.urls'

```

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')]
    },
    {
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'project.wsgi.application'

# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

```

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

*# Static files (CSS, JavaScript, Images)*  
*# <https://docs.djangoproject.com/en/2.2/howto/static-files/>*

STATIC\_URL = '/static/'

LOGIN\_REDIRECT\_URL = 'home'

LOGOUT\_REDIRECT\_URL = 'home'

MARKDOWN\_TAGS = ['h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'b', 'i', 'hr', 'table',  
'thead', 'tbody', 'tr', 'th', 'td', 'strong', 'em', 'tt', 'p', 'pre', 'br',  
'span', 'blockquote', 'code', 'ul', 'ol', 'li', 'dd', 'dt', 'a', 'lt', 'gt']

MARKDOWN\_ATTRS = {'a': ['href', 'alt', 'title'], 'code': ['class']}

PAGINATE\_ENTRIES\_BY = 10

## `\project\urls.py`

`"""project URL Configuration`

`The urlpatterns list routes URLs to views. For more information please see:  
https://docs.djangoproject.com/en/2.2/topics/http/urls/`

`Examples:`

`Function views`

- `1. Add an import: from my_app import views`
- `2. Add a URL to urlpatterns: path('', views.home, name='home')`

`Class-based views`

- `1. Add an import: from other_app.views import Home`
- `2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')`

`Including another URLconf`

- `1. Import the include() function: from django.urls import include, path`
- `2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))`

`"""`

```
from django.contrib import admin
from django.urls import path, include
from django.views.generic.base import TemplateView
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('accounts.urls')),
    path('accounts/', include('django.contrib.auth.urls')),

    path("", include("argupedia.urls")),
]
```



\argupedia\views.py

```
from django.http import Http404
from django.conf import settings
from django.views import View
from .models import Entry
from .models import Scheme
from .models import SchemeStructure
from .models import CriticalQuestion
from django.db.models import Count
from django.core.paginator import EmptyPage, PageNotAnInteger, Paginator
from django.views.generic.detail import DetailView
from django.contrib.auth.models import User
from django.shortcuts import redirect, render
from django.views.decorators.csrf import csrf_exempt
from django.http import HttpResponse
import json

# Create your views here.
class HomeView(View):
    def get(self, request):
        root_nodes = Entry.objects.root_nodes()
        root_nodes = root_nodes.annotate(overall_votes=(Count("upvotes") -
Count("downvotes"))).order_by("-overall_votes")

        page = request.GET.get('page', 1)
        paginator = Paginator(root_nodes, 5)
        try:
            queryset = paginator.page(page)
        except PageNotAnInteger:
            queryset = paginator.page(1)
        except EmptyPage:
            queryset = paginator.page(paginator.num_pages)

        return render(request, "home.html", {"entries": queryset})

class PostsDetailView(View):
    def get(self, request, pk_post):
        subtree = Entry.objects.get(pk=pk_post)
        title = subtree.title
        root_nodes = subtree.get_descendants(include_self=True)
        root_nodes = root_nodes.annotate(overall_votes=(Count("upvotes") -
Count("downvotes"))).order_by("-overall_votes")
        return render(request, "post_detail.html", {"entries": root_nodes, "title"
: title, "root_post_id" : pk_post, "is_parent": (subtree.parent == None)})

class MyDiscussionsView(View):
    def get(self, request, username):
        root_nodes = Entry.objects.root_nodes()
        root_nodes = root_nodes.filter(user=request.user)
        root_nodes = root_nodes.annotate(overall_votes=(Count("upvotes") -
Count("downvotes"))).order_by("-created_date")

        page = request.GET.get('page', 1)
        paginator = Paginator(root_nodes, 5)
        try:
            queryset = paginator.page(page)
        except PageNotAnInteger:
            queryset = paginator.page(1)
```

```

    except EmptyPage:
        queryset = paginator.page(paginator.num_pages)

    return render(request, "home.html", {"entries": queryset})

class ChooseSchemeView(View):
    def get(self, request, pk_post):
        all_schemes = Scheme.objects.all()
        entry = ''
        criticalQuestions = ''
        if(pk_post != 0):
            entry = Entry.objects.get(pk=pk_post)
            attacking = Entry.objects.get(pk=pk_post)
            criticalQuestions =
CriticalQuestion.objects.filter(related_scheme=attacking.scheme_used.id)

        return render(request, "choose_scheme.html", {"schemes": all_schemes,
"pk_post": pk_post, "critiques": criticalQuestions, "entry": entry})

class CreatePost(View):
    def get(self, request, pk_scheme):
        argumentation_scheme =
SchemeStructure.objects.filter(scheme=pk_scheme).order_by('ordering')
        return render(request, "create_post.html", {"scheme_structure":
argumentation_scheme, "scheme_id": pk_scheme})

    def post(self, request, pk_scheme):
        argumentation_scheme =
SchemeStructure.objects.filter(scheme=pk_scheme).order_by('ordering')
        stringBuilder = []
        for tup in argumentation_scheme:
            stringBuilder.append('**')
            stringBuilder.append(tup.section_title)
            stringBuilder.append(':')
            stringBuilder.append('**')
            stringBuilder.append('<br>')
            stringBuilder.append(request.POST.get(tup.section_title, ""))
            stringBuilder.append('<br>')
        baldur = ''.join(stringBuilder)
        entry = Entry(user=request.user, content=baldur,
title=request.POST.get('post_title', ""),
scheme_used=Scheme.objects.get(pk=pk_scheme))
        entry.save()
        entry.upvotes.add(request.user)
        return redirect('posts-detail-view', pk_post=entry.id)

class DeletePost(View):
    def get(self, request, pk_post):
        Entry.objects.get(id=pk_post).delete()
        Entry.objects.rebuild()
        return redirect('home')

@csrf_exempt
def upvote(request):
    entry = Entry.objects.get(id=request.POST.get('pk_post'))
    data = {}
    data['id'] = entry.pk

```

```

entry.downvotes.remove(request.user)
if request.user in entry.upvotes.all():
    entry.upvotes.remove(request.user)
    data['user_upvoted'] = False
else:
    entry.upvotes.add(request.user)
    data['user_upvoted'] = True

data['upvotes'] = entry.upvotes.count()
data['downvotes'] = entry.downvotes.count()

x = json.dumps(data)
return HttpResponse(x, content_type="application/json")

@csrf_exempt
def downvote(request):
    entry = Entry.objects.get(id=request.POST.get('pk_post'))
    data = {}
    data['id'] = entry.pk
    entry.upvotes.remove(request.user)
    if request.user in entry.downvotes.all():
        entry.downvotes.remove(request.user)
        data['user_downvoted'] = False
    else:
        entry.downvotes.add(request.user)
        data['user_downvoted'] = True

    data['upvotes'] = entry.upvotes.count()
    data['downvotes'] = entry.downvotes.count()

    x = json.dumps(data)
    return HttpResponse(x, content_type="application/json")

class CounterPost(View):
    def get(self, request, pk_scheme, pk_post, critical_question_id):
        argumentation_scheme =
        SchemeStructure.objects.filter(scheme=pk_scheme).order_by('ordering')
        attacking = Entry.objects.get(pk=pk_post)
        criticalQuestions =
        CriticalQuestion.objects.filter(related_scheme=attacking.scheme_used.id)
        return render(request, "counter_post.html", {"scheme_structure":
        argumentation_scheme, "scheme_id": pk_scheme, "critical_question_id":
        critical_question_id, "entry": Entry.objects.get(pk=pk_post), "critiques":
        criticalQuestions})

    def post(self, request, pk_scheme, pk_post, critical_question_id):
        argumentation_scheme =
        SchemeStructure.objects.filter(scheme=pk_scheme).order_by('ordering')
        stringBuilder = []
        critique = CriticalQuestion.objects.get(pk=request.POST.get('critique'))
        stringBuilder.append('Critique Position: *')
        stringBuilder.append(critique.question)
        stringBuilder.append('*')
        stringBuilder.append('<br>')
        for tup in argumentation_scheme:
            stringBuilder.append('**')
            stringBuilder.append(tup.section_title)
            stringBuilder.append(':')

```

```

        stringBuilder.append('**')
        stringBuilder.append('<br>')
        stringBuilder.append(request.POST.get(tup.section_title, ""))
        stringBuilder.append('<br>')
        baldur = ''.join(stringBuilder)
        attacking = Entry.objects.get(pk=pk_post)

        entry = Entry(user=request.user, content=baldur,
title=request.POST.get('post_title', ""),
scheme_used=Scheme.objects.get(pk=pk_scheme), critical_question=critique,
parent=attacking)
        entry.save()
        entry.upvotes.add(request.user)
        return redirect('posts-detail-view', pk_post=pk_post)

class VisualizeView(View):
    def get(self, request, pk_post):
        subtree = Entry.objects.get(pk=pk_post)
        root_nodes = subtree.get_descendants(include_self=True)
        set_in = set() #set in
        set_out = set() #set out
        set_undec = set() #set undecidable
        for entry in root_nodes:
            if (entry.has_children == False and entry.is_root_node() == False):
                if (entry.critical_question.is_attack_on_conclusion == False):
                    set_in.add(entry.pk)
        while(True):
            prev_set_in = set_in.copy()
            prev_set_out = set_out.copy()
            for entry in root_nodes:
                if entry.pk not in set_in and entry.pk not in set_out:
                    okay = True
                    for y in entry.get_attackers():
                        if y.pk not in set_out:
                            okay = False
                            break
                    if okay == True:
                        set_in.add(entry.pk)
            for entry in root_nodes:
                if entry.pk not in set_out and entry.pk not in set_in:
                    okay = False
                    for y in entry.get_attackers():
                        if y.pk in set_in:
                            okay = True
                            break
                    if okay == True:
                        set_out.add(entry.pk)
            if(prev_set_in == set_in and prev_set_out == set_out):
                break
        for entry in root_nodes: #append unlabelled sets to set undecidable
            if entry.pk not in set_in and entry.pk not in set_out:
                set_undec.add(entry.pk)
        #print(set_in)
        #print(set_out)
        #print(set_undec)
        return render(request, "visualize.html", {"post_id": pk_post, "entries":
root_nodes, "set_in": set_in, "set_out": set_out, "set_undec" : set_undec})

```

\argupedia\urls.py

```
from django.contrib.auth.decorators import user_passes_test
from django.urls import path
from django.views.generic.base import TemplateView

from .views import HomeView, VisualizeView
from .views import PostsDetailView
from .views import MyDiscussionsView
from .views import ChooseSchemeView
from .views import CreatePost
from .views import DeletePost
from .views import CounterPost
from argupedia import views

logged_users_redirect = user_passes_test(lambda u: u.is_anonymous, "/")

urlpatterns = [
    path('', HomeView.as_view(), name='home'),
    path("posts/<int:pk_post>/", PostsDetailView.as_view(), name="posts-detail-
view"),
    path("posts/delete/<int:pk_post>/", DeletePost.as_view(), name="posts-
delete"),
    path("posts/mydiscussions/<str:username>/", MyDiscussionsView.as_view(),
name="my-discussions-view"),
    path("posts/create/choose_scheme/<int:pk_post>/", ChooseSchemeView.as_view(),
name="choose-scheme-view"),

    path("posts/create/input_argument/<str:pk_scheme>/", CreatePost.as_view(),
name="create-post-view"),
    #path("posts/counter/<str:pk_scheme>/<int:pk_post>/", CounterPost.as_view(),
name="counter-post-view"),

    path("posts/counter/<str:pk_scheme>/<int:pk_post>/<int:critical_question_id>/",
CounterPost.as_view(), name="counter-post-view"),
    #visualization
    path("posts/visualize/<int:pk_post>/", VisualizeView.as_view(),
name="visualize-view"),

    path("posts/upvote/", views.upvote, name="upvote"),
    path("posts/downvote/", views.downvote, name="downvote"),
]
```

\argupedia\models.py

```
from django.db import models
import re
import bleach
import markdown
from django.conf import settings
from django.contrib.auth.models import User
from django.contrib.contenttypes.fields import GenericForeignKey
from django.contrib.contenttypes.models import ContentType
from django.urls import reverse
from django.utils import timezone
from django.utils.functional import cached_property
from django.utils.html import format_html
from mptt.models import MPTTModel, TreeForeignKey

class Scheme(models.Model):
    scheme_name = models.CharField(max_length=100)
    def __str__(self):
        return self.scheme_name

class SchemeStructure(models.Model):
    ORDERING = [('1', '1'), ('2', '2'), ('3', '3'), ('4', '4'), ('5', '5'), ('6', '6'), ('7', '7'), ('8', '8'),]
    scheme = models.ForeignKey(Scheme, on_delete=models.CASCADE, related_name="scheme")
    section_title = models.CharField(max_length=100)
    ordering = models.CharField(max_length=50, choices=ORDERING)
    section_description = models.CharField(max_length=100, blank=True)

    def __str__(self):
        return self.section_title

class CriticalQuestion(models.Model):
    related_scheme = models.ForeignKey(Scheme, on_delete=models.CASCADE, related_name="related_scheme")
    is_attack_on_conclusion = models.BooleanField()
    question = models.CharField(max_length=500)

    def __str__(self):
        return self.question

class Entry(MPTTModel):
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name="author")
    parent = TreeForeignKey("self", null=True, blank=True, on_delete=models.CASCADE, related_name="children")
    content = models.TextField(max_length=4000, default="", help_text="")
    content_formatted = models.TextField(default="")
    title = models.TextField(max_length=100, default="", help_text="", blank=True)
    upvotes = models.ManyToManyField(User, blank=True, related_name="upvotes")
    downvotes = models.ManyToManyField(User, blank=True, related_name="downvotes")
    created_date = models.DateTimeField(default=timezone.now)
    modified_date = models.DateTimeField(blank=True, null=True)
    scheme_used = models.ForeignKey(Scheme, on_delete=models.CASCADE, related_name="scheme_used")
    critical_question = models.ForeignKey(CriticalQuestion, null=True, blank=True, on_delete=models.CASCADE, related_name="critical_question")
```

```

class MPTTMeta:
    order_insertion_by = ["-created_date"]

class Meta:
    verbose_name_plural = "Entries"

def __str__(self):
    if len(self.content) > 45:
        return f'"{self.content:.45}..."'
    return f'"{self.content}"'

def get_absolute_url(self):
    return reverse('entry-detail-view', args=[str(self.id)])

def format_content(self):
    self.content_formatted = self.content
    # Clean and format content
    self.content_formatted =
bleach.clean(markdown.markdown(self.content_formatted,
extensions=["extra"]), settings.MARKDOWN_TAGS, settings.MARKDOWN_ATTRS, )
    self.content = bleach.clean(self.content, settings.MARKDOWN_TAGS,
settings.MARKDOWN_ATTRS)

def get_attackers(self):
    attackers = set()
    if len(self.get_children()) != 0:
        for item in self.get_children():
            attackers.add(item)
    if (self.parent != None):
        if self.critical_question.is_attack_on_conclusion == True:
            attackers.add(self.parent)
    return attackers

def save(self, *args, **kwargs):
    created = True if not self.pk else False
    if not created:
        self.modified_date = timezone.now()
    self.format_content()
    super().save(*args, **kwargs)
    if created:
        self.upvotes.add(self.user)

@cached_property
def votes_sum(self):
    """
    Returns subtract of downvotes from upvotes
    """
    return self.upvotes.count() - self.downvotes.count()

@cached_property
def root_pk(self):
    """
    Returns id of root node (if node is a root node then root_pk = self.pk)
    """
    return self.get_root().pk

@cached_property
def has_children(self):

```

```

    """
    Returns True if entry has children nodes
    """
    return self.get_children().exists()

def parent_formatted(self):
    """
    Formats parent node into a hyperlink
    """
    if not self.parent:
        return "-"
    url = reverse("admin:app_entry_change", args=(self.parent.pk,))
    return format_html('<a href="{}">#{}</a>', url, self.parent.pk)

def user_formatted(self):
    """
    Formats author into a hyperlink
    """
    url = reverse("admin:app_user_change", args=(self.user.pk,))
    return format_html('<a href="{}">{</a>', url, self.user.username)

parent_formatted.short_description = "Parent entry"
user_formatted.short_description = "User"

```



\argupedia\admin.py

```
from django.contrib import admin
from .models import Entry
from .models import Scheme
from .models import SchemeStructure
from .models import CriticalQuestion
# Register your models here.
admin.site.register(Entry)
admin.site.register(Scheme)
admin.site.register(SchemeStructure)
admin.site.register(CriticalQuestion)
```

\argupedia\templates\choose\_scheme.html

```
{% extends 'base.html' %}
{% block title %}Choose Scheme{% endblock %}
{% block content %}
    {% if pk_post == 0 %}
        <div class="container-fluid">
            <div class="row justify-content-center my-3">
                <h5>Select Argument Scheme</h5>
            </div>
            {% for scheme in schemes %}
                <div class="row justify-content-center">
                    <a href="{% url 'create-post-view' pk_scheme=scheme.pk %}"
class="btn btn-lg btn-outline-dark mb-2" style="width: 500px">{{
scheme.scheme_name }}</a>
                </div>
            {% endfor %}
        </div>
    {% else %}
        <div class="container-fluid">
            <div class="row">
                <div class="col-lg-6">
                    <div class="row justify-content-center">
                        <h5>Original Argument</h5>
                    </div>
                    <div class="row justify-content-center ml-5">
                        {{ entry.content_formatted|safe }}
                    </div>
                </div>

                <div class="col-lg-6">

                    <div class="container center_div">
                        <h4 style="margin-bottom: 15px; margin-top: 10px;text-align: center;">Counter Argument</h4>
                        <label for="text" style="margin-left: 15px;"><strong>Critique Position</strong></label>
                        <div class="col-md-12">
                            <select class="form-control" id="critique"
name="critique" required onchange="configureVisibility();appendLinks();">
                                <option value="">Select</option>
                                {% for critique in critiques %}
                                    <option
value="{{critique.id}}">{{critique.question}}</option>
                                {% endfor %}
                            </select>
                        </div>
                        <div class="col-md-4" id="scheme_list">
                            <div class="row justify-content-center my-3">
                                <h5>Choose Counter-Argument Scheme</h5>
                            </div>
                            {% for scheme in schemes %}
                                <div class="row justify-content-center">
                                    <a href="#" class="btn btn-outline-dark btn-lg
mb-2" id="btn{{scheme.pk}}" style="width: 400px">
                                        {{ scheme.scheme_name }}
                                    </a>
                                </div>
                            {% endfor %}
                        </div>
                    </div>
                </div>
            </div>
        </div>
    {% endif %}
{% endblock %}
```

```

        </div>
    </div>
</div>
</div>

<script>
    function appendLinks()
    {
        {% for scheme in schemes %}
            var url = "{% url 'counter-post-view' pk_scheme=scheme.pk
pk_post=entry.id critical_question_id=123 %}"
.replace('123',
$('#critique').val());
            document.getElementById("btn" +
{{scheme.pk}}).setAttribute('href', url.toString());
        {% endfor %}
    }
    function configureVisibility()
    {
        if($("#critique").val()=="")
        {
            document.getElementById("scheme_list").style.visibility =
"hidden";
        }
        else
        {
            document.getElementById("scheme_list").style.visibility =
"visible";
        }
    }
    configureVisibility();
</script>
{% endif %}

{% endblock %}

```

\argupedia\templates\counter\_post.html

```
{% extends 'base.html' %}
{% block title %}Counter Argument{% endblock %}
{% block content %}

    <div class="container-fluid">
        <div class="row">
            <div class="col-md-5 centered" style="padding-bottom : 40px">
                <div class="row justify-content-center my-3">
                    <h5>Original Argument</h5>
                </div>
                <div class="row justify-content-center my-3">
                    {{ entry.content_formatted|safe }}
                </div>
            </div>

            <div class="col-md-7 centered" style="padding-bottom : 40px">
                <div class="form-area">
                    <form action="{% url 'counter-post-view' pk_scheme=scheme_id
pk_post=entry.id critical_question_id=critical_question_id %}" method="post">
                        {% csrf_token %}
                        <div class="container center_div pl-3">
                            <h4 style="margin-bottom: 15px; margin-top: 10px;text-align: center;">Provide your counter-argument</h4>
                            <label for="text"><strong>Critique
Position:</strong></label>
                            <div class="col-md-12">
                                <select class="form-control" id="critique"
name="critique" required>
                                    <option value="">Select</option>
                                    {% for critique in critiques %}
                                        <option value="{{critique.id}}" {% if
critical_question_id == critique.id %}selected{% endif
%}>{{critique.question}}</option>
                                    {% endfor %}
                                </select>
                            </div>
                            {% for structure in scheme_structure %}
                                <div class="form-group">
                                    <label for="text"><strong>{{
structure.section_title }}:</strong></label>
                                    <div class="col-md-12">
                                        <textarea class="form-control" name="{{
structure.section_title }}" id="{{ structure.section_title }}" rows="2"
placeholder="{{ structure.section_description }}" required></textarea>
                                    </div>
                                </div>
                            {% endfor %}
                            <input type="submit" name="submit" class="btn btn-md
active" style="background-color:#2d3246; color :white; width:120px; padding-
left:10px;padding-bottom:5px;margin-left:270px;" role="button" aria-pressed="true"
value="Submit"/>
                            <a href="{% url 'home' %}" class="btn btn-md active"
style="background-color:#2d3246; color :white; width:120px; padding-
left:10px;padding-bottom:5px;">Cancel</a>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
{% endblock %}
```

```
        </div>
      </div>
    </div>
{% endblock %}
```

```

\argupedia\templates\create_post.html

{% extends 'base.html' %}
{% block title %}Choose Debate Topic{% endblock %}
{% block content %}

    <div class="container center_div">
        <div class="col-md-12 centered" style="padding-bottom : 40px">
            <div class="form-area">

                <form action="{% url 'create-post-view' pk_scheme=scheme_id %}"
method="post">

                    {% csrf_token %}
                    <div class="container center_div pl-3">
                        <h3 style="margin-bottom: 15px; margin-top: 10px;text-align: center;">Provide your argument</h3>
                        <label for="text"><strong>Topic Title:</strong></label>
                        <div class="col-md-12">
                            <input type="text" class="form-control" id="post_title"
name="post_title" placeholder="Title" value="" required>
                        </div>

                        {% for structure in scheme_structure %}
                        <div class="form-group">
                            <label for="text"><strong>{{ structure.section_title
}}:</strong></label>

                            <div class="col-md-12">
                                <textarea class="form-control" name="{{
structure.section_title }}" id="{{ structure.section_title }}" rows="2"
placeholder="{{ structure.section_description }}" required></textarea>
                            </div>
                        </div>
                        {% endfor %}

                        <input type="submit" name="submit" class="btn btn-outline-
dark" style="width:120px; padding-left:10px;padding-bottom:5px;margin-left:400px;"
role="button" aria-pressed="true" value="Submit"/>
                        <a href="{% url 'home' %}" class="btn btn-outline-dark"
style="width:120px; padding-left:10px;padding-bottom:5px;">Cancel</a>
                    </div>
                </form>
            </div>
        </div>
    </div>
{% endblock %}

```

\argupedia\templates\home.html

```
{% extends 'base.html' %}
{% block title %}Home{% endblock %}
{% block content %}

    <div class="container-fluid">
        <section class="search-banner bg-light text-black py-4 bg-white text-dark"
id="banner">
            <div class="row" >
                <div class="col-lg-7">
                    <div class="row ml-2" >
                        <h5 class="pl-3"><strong>Topics</strong></h5>
                    </div>
                    <div class="row ml-2" >
                        {% if not user.is_authenticated %}
                            <h7 class="pl-3"><a href="/accounts/login/">Login</a>
or <a href="/accounts/signup">Register</a> to create and engage in debates!</h7>
                        {% endif %}
                    </div>
                </div>
                {% if user.is_authenticated %}
                    <div class="col-lg-4 text-right">
                        <div class="btn-group">
                            <a href="{% url 'choose-scheme-view' pk_post=0 %}"
class="btn btn-outline-dark">Create a Debate Topic</a>
                        </div>
                    </div>
                {% endif %}
            </div>
        </section>
    </div>

    {% load mptt_tags %}
    <ul class="entry" style="padding-left: 38px;">
        {% if not entries %}
            <h6>Oops. There are no discussions to show.</h6>
        {% endif %}
        <!-- mttpt tree start -->
        {% for node in entries %}
            {% if node.user.pk == user.pk %}
                <li class="usernode">
                    {% elif node.is_root_node %}
                <li class="rootnode">
                    {% elif node.user.pk == node.get_root.user.pk %}
                <li class="opnode">
                    {% else %}
                <li class="commentnode">
            {% endif %}
            <div class="entryBox">
                <a class="entry-anchor" id="{{ node.pk }}"></a>
                <p style="margin-bottom: 1px">
                    <a href="{% url 'posts-detail-view' node.pk %}" style="margin-
left: 5px; font-size: 14px;"><strong>{{ node.title }}</strong></a>
                    <small><em>Votes:</em></small>
                    <strong class="entry-points" data-id="{{ node.pk }}">{{
node.votes_sum }}</strong>
                </p>
                <div id="content{{ node.pk }}" class="entry-content {% if node.deleted
%}>deleted text-muted{% endif %}" value="{{ node.pk }}" style="{% if entry.pk ==
```

```

node.pk %}background-color: yellow;{% endif %}">{{ node.content_formatted|safe
}}</div>
</div>
</li>
{% if node.is_root_node %}
    <hr>
{% endif %}
{% endfor %}
</ul>

<div class="pagination pb-3">
<span class="step-links">
    {% if entries.has_previous %}
        <a href="?page=1" class="btn btn-sm btn-outline-dark">&laquo;
First</a>
        <a href="?page={{ entries.previous_page_number }}" class="btn btn-sm
btn-outline-dark">Previous</a>
    {% endif %}

    <span class="current"><em>Page {{ entries.number }} of {{
entries.paginator.num_pages }}</em></span>

    {% if entries.has_next %}
        <a href="?page={{ entries.next_page_number }}" class="btn btn-sm btn-
outline-dark">Next</a>
        <a href="?page={{ entries.paginator.num_pages }}" class="btn btn-sm
btn-outline-dark">Last &raquo;</a>
    {% endif %}
</span>
</div>

{% endblock %}

```



\argupedia\templates\post\_detail.html

**Note:** The code in this file was inspired from the project at the following source:

**Source:** <https://github.com/MakuZo/bloggy>

```
{% extends 'base.html' %}
{% block title %}Post Detail{% endblock %}
{% block content %}

    <div class="container-fluid">
        <section class="search-banner bg-light text-black pt-3 bg-white text-dark"
id="banner">
            <div class="row" >
                <div class="col-lg-7">
                    <div class="row ml-2" >
                        <h5 class="pl-3">{{title}}</h5>
                    </div>
                    <div class="row ml-2" >
                        {% if not user.is_authenticated %}
                            <h7 class="pl-3"><a href="/accounts/login/">Login</a>
or <a href="/accounts/signup/">Register</a> to contribute to debate!</h7>
                        {% endif %}
                    </div>
                </div>
                {% if is_parent %}
                    <div class="col-lg-4 text-right">
                        <div class="btn-group">
                            <a href="{% url 'visualize-view' root_post_id %}"
target="_blank" class="btn btn-outline-dark">Visualize</a>
                        </div>
                    </div>
                {% endif %}
            </div>
        </section>
    </div>

    {% load mptt_tags %}
    <ul class="entry" style="padding-left: 38px;">
        {% if not entries %}
            <h6>Oops. There are no discussions to show.</h6>
        {% endif %}
        <!-- mptt tree start -->
        {% recursetree entries %}
            {% if node.user.pk == user.pk %}
                <li class="usernode">
                    {% elif node.is_root_node %}
                <li class="rootnode">
                    {% elif node.user.pk == node.get_root.user.pk %}
                <li class="opnode">
                    {% else %}
                <li class="commentnode">
            {% endif %}
            <div class="entryBox">
                <a class="entry-anchor" id="{{ node.pk }}"></a>
                <p style="margin-bottom: 1px">
                    <a href="{% url 'posts-detail-view' node.pk %}" style="margin-
left: 5px; font-size: 14px;"><strong>{{ node.user.username }}</strong></a>
                    <small><em>Votes:</em></small>
                    <strong class="entry-points" data-id="{{ node.pk }}">{{
```

```

node.votes_sum }}</strong>
    {% if user.is_authenticated %}
        {% if not node.deleted %}
            <button class="upvote" {% if user in node.upvotes.all
}%}user-upvoted{% endif %}" data-id="{{node.pk}}">&#43;</button>
            <button class="downvote" {% if user in node.downvotes.all
}%}user-downvoted{% endif %}" data-id="{{node.pk}}">&#8722;</button>
        {% endif %}
        <small><em>{{node.created_date}}</em></small>
        <a href="{% url 'choose-scheme-view' node.pk %}"
class="answerButton" value="{{node.pk}}" style="font-size: 12px">counter</a>
{% comment %}...
            {% if node.user.pk == user.pk and not node.deleted %}
                <a href="{% url 'posts-delete' node.pk %}"
class="deleteButton" value="{{node.pk}}" style="font-size: 12px; color: red;
visibility: none;">delete</a>
            {% endif %}
{% endcomment %}
        {% endif %}
    </p>
    <div id="content{{node.pk}}" class="entry-content" {% if node.deleted
}%}deleted text-muted{% endif %}" value="{{node.pk}}" style="{% if entry.pk ==
node.pk %}background-color: yellow;{% endif %}">{{node.content_formatted|safe
}}</div>
</div>

    {% if node.has_hidden_children %}
        <a style="margin-left: 15px; font-size: 12px; background-color:
lightblue; color: white;" href="{% url 'posts-detail-view' node.pk %}">Continue
into this discussion -></a>
    {% endif %}
    {% if not node.is_leaf_node %}
        {% if node.is_root_node %}
            <div style="margin-bottom: 12px;"></div>
        {% endif %}
        <ul class="children entry" style="padding-left: 25px">
            {{ children }}
        </ul>
    {% endif %}
</li>
    {% if node.is_root_node %}
        <hr>
    {% endif %}
{% endrecursetree %}
</ul>

<script>
    $(document).on('click', '.upvote', function (e) {
        e.preventDefault();
        $.ajax({
            url: "{% url 'upvote' %}",
            type: 'POST',
            data: {'pk_post': e.target.dataset['id']},
            success: function (data) {
                $('<div>downvoted">
                    if(data.user_upvoted == true)
                    {
                        e.target.classList.add("user-upvoted")

```

```

        } else {
            e.target.classList.remove("user-upvoted")
        }
        $(`.entry-points[data-id='${data.id}']`)[0].innerHTML =
data.upvotes - data.downvotes
    },
    });
});

$(document).on('click', '.downvote', function (e) {
    e.preventDefault();
    $.ajax({
        url: "{% url 'downvote' %}",
        type: 'POST',
        data: {'pk_post': e.target.dataset['id']},
        success: function (data) {
            $`.upvote[data-id='${data.id}']`.removeClass("user-upvoted")
            if(data.user_downvoted == true){
                e.target.classList.add("user-downvoted")
            } else {
                e.target.classList.remove("user-downvoted")
            }
            $`.entry-points[data-id='${data.id}']`)[0].innerHTML =
data.upvotes - data.downvotes
        },
    });
});

</script>
{% endblock %}

```

\argupedia\templates\visualize.html

*Note: The code in this file was inspired from the project at the following source:*

*Source: <http://www.d3noob.org/2013/03/d3js-force-directed-graph-example-basic.html>*

```
{% extends 'base.html' %}
{% block title %}Visualization{% endblock %}
{% block content %}
    {% load static %}

    <script src="https://d3js.org/d3.v4.min.js" type="text/javascript"></script>
    <script src="https://d3js.org/d3-selection-multi.v1.js"></script>

    <div class="container" style="height: 650px; width: 1000px;">
        <svg width="960" height="600"></svg>
        <script src="https://d3js.org/d3.v4.min.js"
type="text/javascript"></script>
        <script src="https://d3js.org/d3-selection-multi.v1.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/d3-tip/0.7.1/d3-
tip.min.js"></script>

        <style>

        </style>
        <script type="text/javascript">
            {% load mptt_tags %}
            var gr = {
                nodes: [
                    {% recursetree entries %}
                    {
                        id: {{node.pk}}, argument:
'{{node.content_formatted|safe}}', votes: '{{node.votes_sum}}', node_text: {% if
node.parent == null %}"S"{% else %}""{% endif %}, color:{% if node.pk in set_in
%}"#CBF6A9"{% elif node.pk in set_out %}"#F6B6A9"{% else %}"#EFF1EE"{% endif %}
                    },
                    {{ children }}
                {% endrecursetree %}
            ],
            legendVals: [
                {legend_text: "In", legend_color:"#CBF6A9" },
                {legend_text: "Out", legend_color:"#F6B6A9" },
                {legend_text: "Undec", legend_color:"#EFF1EE" },
            ],
            links: [
                {% recursetree entries %}
                {% if node.parent != null %}
                {
                    source: {{ node.pk }}, target: {{ node.parent.pk
}}, critique: '{{ node.critical_question.question }}', scheme: '{{
node.scheme_used.scheme_name }}'
                },
                {% if node.critical_question.is_attack_on_conclusion
%}
                {
                    source: {{ node.parent.pk }}, target: {{
node.pk }}, critique: 'Backed by premise claims', scheme: '{{
```

```

node.parent.scheme_used.scheme_name }}'
        },
        {% endif %}
    {% endif %}
    {{ children }}
    {% endrecursetree %}

    ]
};

var tip = d3.tip()
    .attr('class', 'd3-tip')
    .offset([-10, 0]);

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    node,
    link;

svg.call(tip);

svg.append('defs').append('marker')
    .attr({ 'id': 'arrowhead',
        'viewBox': '-0 -5 10 10',
        'refX': 16,
        'refY': -0.5,
        'orient': 'auto',
        'markerWidth': 2.2,
        'markerHeight': 10,
        'xoverflow': 'visible' })
    .append('svg:path')
    .attr('d', 'M 0,-5 L 10 ,0 L 0,5')
    .attr('fill', '#999')
    .style('stroke', 'none');

var simulation = d3.forceSimulation()
    .force("link", d3.forceLink().id(function (d) {return
d.id;}).distance(100).strength(0.5))
    .force("charge", d3.forceManyBody().strength(-400))
    .force("x", d3.forceX())
    .force("y", d3.forceY())
    .force("center", d3.forceCenter(width / 2, height / 2));

update(gr.links, gr.nodes, gr.legendVals);

function update(links, nodes, legendVals) {
    link = svg.selectAll(".link")
        .data(links)
        .enter()
        .append("line")
        .attr("class", "link")
        .style("stroke-width", 1)

```

```

edgepaths = svg.selectAll(".edgepath")
    .data(links)
    .enter()
    .append('path')
    .attr('marker-end', 'url(#arrowhead)')
    .attrs({
        'class': 'edgepath',
        'fill-opacity': 0,
        'stroke-opacity': 1,
        'stroke-width': 6,
        'stroke': function (d) {return '#e0e0e0'},
        'id': function (d, i) {return 'edgepath' + i}
    })
    .on('mouseover', function (d, i)
{tip.html(d.critique);tip.show(); d3.select(this).attr("stroke", "#cdcdcd");})
    .on('mouseout', function (d, i)
{tip.hide();d3.select(this).attr("stroke", "#e0e0e0");});

edgelabels = svg.selectAll(".edgelabel")
    .data(links)
    .enter()
    .append('text')
    .style("pointer-events", "none")
    .attrs({
        'class': 'edgelabel',
        'id': function (d, i) {return 'edgelabel' + i},
        'font-size': 9,
        'fill': 'black'
    });

edgelabels.append('textPath')
    .attr('xlink:href', function (d, i) {return '#edgepath' + i})
    .style("text-anchor", "middle")
    .style("pointer-events", "none")
    .style("font-weight", "bold")
    .attr("startOffset", "50%")
    .text(function (d) {return d.scheme});

node = svg.selectAll(".node")
    .data(nodes)
    .enter()
    .append("g")
    .attr("class", "node")
    .on("click", function (d) {window.location.replace("/posts/" +
d.id + "/" );})
    .on('mouseover', function (d)
{tip.html(d.argument);tip.show();})
    .on('mouseout', tip.hide)
    .call(d3.drag()
        .on("start", dragstarted)
        .on("drag", dragged)
    );

node.append("circle")
    .attr("r", 10)
    .style("fill", function (d) {return d.color;})
    .style("stroke-opacity",0.3)

```

```

        .style("stroke-width", 2);

node.append("title")
    .text(function (d) {return ("vote count: " + d.votes)});
//put name over node here
node.append("text")
    .attr("dy", 4)
    .attr("dx", -15)
    .text(d => "");
//put text over node here
node.append("text")
    .attr("dy", 4)
    .attr("dx", -4)
    .text(function (d) {return d.node_text;});

var legend = svg.selectAll('.legend')
    .data(legendVals)
    .enter()
    .append('g')
    .attr('class', 'legend')
    .attr("transform", (d, i) => `translate(${700},${i * 20})`);

legend.append('rect')
    .attr('width', 15)
    .attr('height', 15)
    .style('fill', (d, i) => d.legend_color)
    .style('stroke', (d, i) => d.legend_color);

legend.append('text')
    .attr('x', 22)
    .attr('y', 12)
    .style("font-size", "12px")
    .text(function(d) { return d.legend_text; });

simulation
    .nodes(nodes)
    .on("tick", ticked);

simulation.force("link")
    .links(links);
}

function ticked() {
    link
        .attr("x1", function (d) {return d.source.x;})
        .attr("y1", function (d) {return d.source.y;})
        .attr("x2", function (d) {return d.target.x;})
        .attr("y2", function (d) {return d.target.y;});

    node.attr("transform", function (d) {return "translate(" + d.x +
        ", " + d.y + ")"});

    edgepaths.attr("d", function (d) {
        var dx = d.target.x - d.source.x,
            dy = d.target.y - d.source.y,
            dr = Math.sqrt(dx * dx + dy * dy);
        return "M" + d.source.x + "," + d.source.y + "A" + dr + "," +
        dr + " 0 0,1 " + d.target.x + "," + d.target.y;
    });

```

```

    });

    edgelabels.attr('transform', function (d) {
        if (d.target.x < d.source.x) {
            var bbox = this.getBBox();

            rx = bbox.x + bbox.width / 2;
            ry = bbox.y + bbox.height / 2;
            return 'rotate(180 ' + rx + ' ' + ry + ')';
        }
        else {
            return 'rotate(0)';
        }
    });
}

function calcTranslationExact(targetDistance, point0, point1)
{
    var x1_x0 = point1.x - point0.x,
        y1_y0 = point1.y - point0.y,
        x2_x0, y2_y0;
    if (y1_y0 === 0) {
        x2_x0 = 0;
        y2_y0 = targetDistance;
    } else {
        var angle = Math.atan((x1_x0) / (y1_y0));
        x2_x0 = -targetDistance * Math.cos(angle);
        y2_y0 = targetDistance * Math.sin(angle);
    }
    return {
        dx: x2_x0,
        dy: y2_y0
    };
}

function dragstarted(d) {
    if (!d3.event.active) simulation.alphaTarget(0.3).restart()
    d.fx = d.x;
    d.fy = d.y;
}

function dragged(d) {
    d.fx = d3.event.x;
    d.fy = d3.event.y;
}

</script>
</div>
{% endblock %}

```



\argupedia\static\argupedia\styles.css

*Note: The code in this file was taken from the following source except*

*Source: <https://github.com/MakuZo/bloggy>*

```
.d3-tip {
  line-height: 1;
  font-size: 11px !important;
  padding: 12px;
  background: rgba(0, 0, 0, 0.8);
  color: #fff;
  border-radius: 2px;
}

.d3-tip:after {
  box-sizing: border-box;
  display: inline;
  font-size: 10px;
  width: 100%;
  line-height: 1;
  color: rgba(0, 0, 0, 0.8);
  content: "\25BC";
  position: absolute;
  text-align: center;
}

.d3-tip.n:after {
  margin: -1px 0 0 0;
  top: 100%;
  left: 0;
}

p {
  display: block;
  word-break: break-all;
  word-wrap: break-word;
  width: 800px;
}

.entry {
  list-style: none;
}

.node {
  font-size: 12px;
}

.node.positive {
  color: green;
}

.node.negative {
  color: red;
}
```

```

.node.user-upvoted {
  color: white;
  background-color: green;

  padding: 1px;
  border-radius: 5px;
  border: none;
}

.node.user-downvoted {
  color: white;
  background-color: red;

  padding: 1px;
  border-radius: 5px;
  border: none;
}

.downvote {
  border-radius: 5px;
  background-color: transparent;
  font-size: 14px;
  font-weight: bold;
  border: none;
}

.upvote {
  border-radius: 5px;
  background-color: transparent;
  font-size: 14px;
  font-weight: bold;
  border: none;
}

.downvote.user-downvoted {
  color: lightcoral;
  background-color: rgba(119, 0, 0, 0.2);
  border: none;
}

.upvote.user-upvoted {
  color: green;
  background-color: rgba(1, 90, 8, 0.2);
  border: none;
}

.entry-anchor {
  display: block;
  position: relative;
  top: -250px;
  visibility: hidden;
}

.entry-points {
  font-size: 12px;
}

 usernode {
  /* border-left: 1px solid rgba(8, 189, 147, 0.666); */

```

```

    border-left: 1px solid rgba(218, 44, 215, 0.673);
}

 usernode p{
    margin-bottom: 5px;
}

 rootnode p{
    margin-bottom: 5px;
}

 opnode {
    border-left: 1px solid rgba(15, 191, 255, 0.496);
}
 commentnode {
    border-left: 1px solid rgba(0, 0, 0, 0.1);
}

/* Custom Markdown Styling */
.entry-content {
    margin-left: 5px;
    word-wrap: break-word;
}
.entry-content h1{
    font-size: 18px;
}

.entry-content h2{
    font-size: 16px;
}

.entry-content h3{
    font-size: 16px;
}

.entry-content h4{
    font-size: 16px;
}

.entry-content h5{
    font-size: 16px;
}

.entry-content h6{
    font-size: 16px;
}

.entry-content pre code {
    background-color: rgb(246, 246, 246);
}

```