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

[A.1 External Libraries used 3](#_Toc38230284)

[A.2 Project Running Instructions 5](#_Toc38230285)

[A.3 Testing Screenshots 6](#_Toc38230286)

[A.4 Code Listing 23](#_Toc38230287)

# A.1 External Libraries used

External libraries used in the project (except the Django framework itesef) 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 test 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:

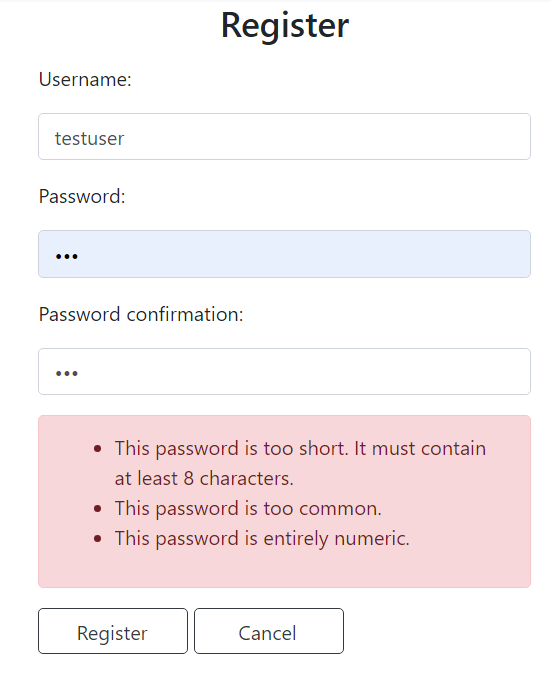
<https://github.com/MakuZo/bloggy>

# A.2 Project Running Instructions

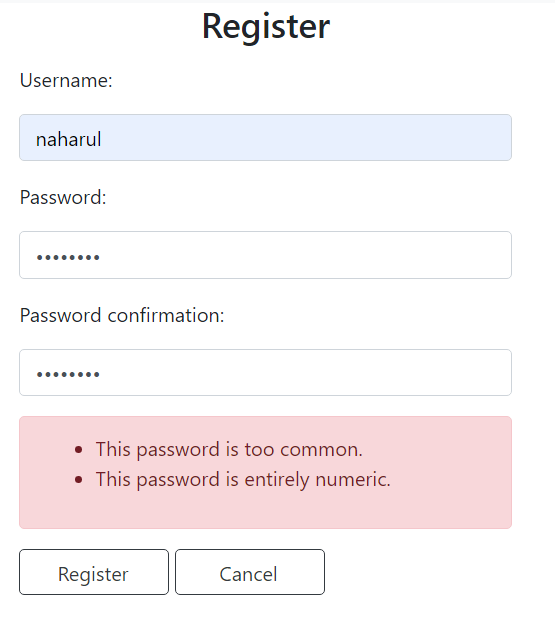
* Extract the project.
* Navigate to the path of the project using command prompt:
  + cd <project location>
* Navigate to the scripts folder using command prompt:
  + cd Scripts
* Activate virtual environment:
  + activate.bat
* Back out of the folder into the root folder:
  + cd ..
* Run the following 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

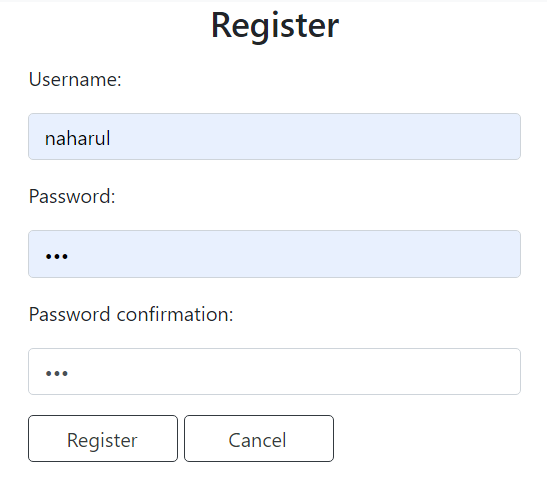
### A.3.1

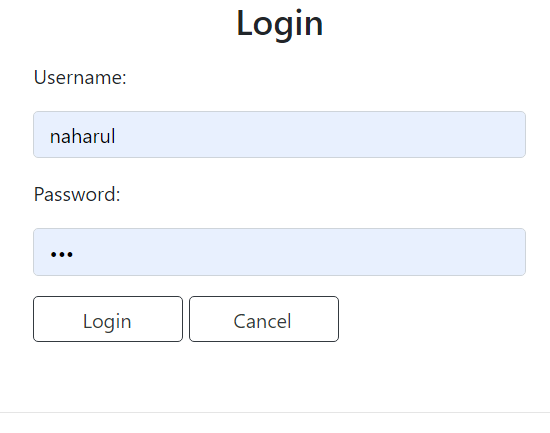


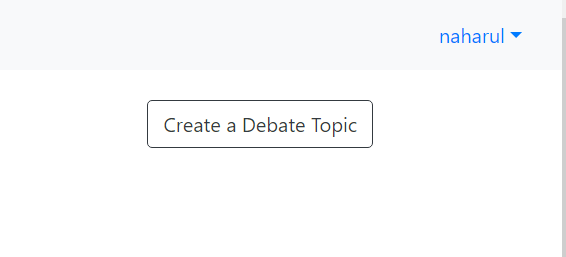
### A.3.2



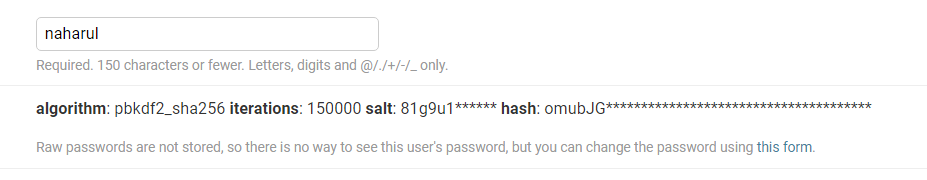
### A.3.3





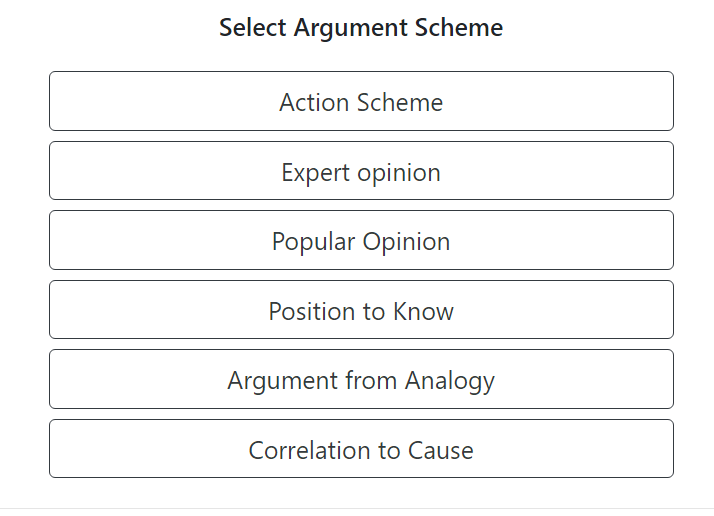


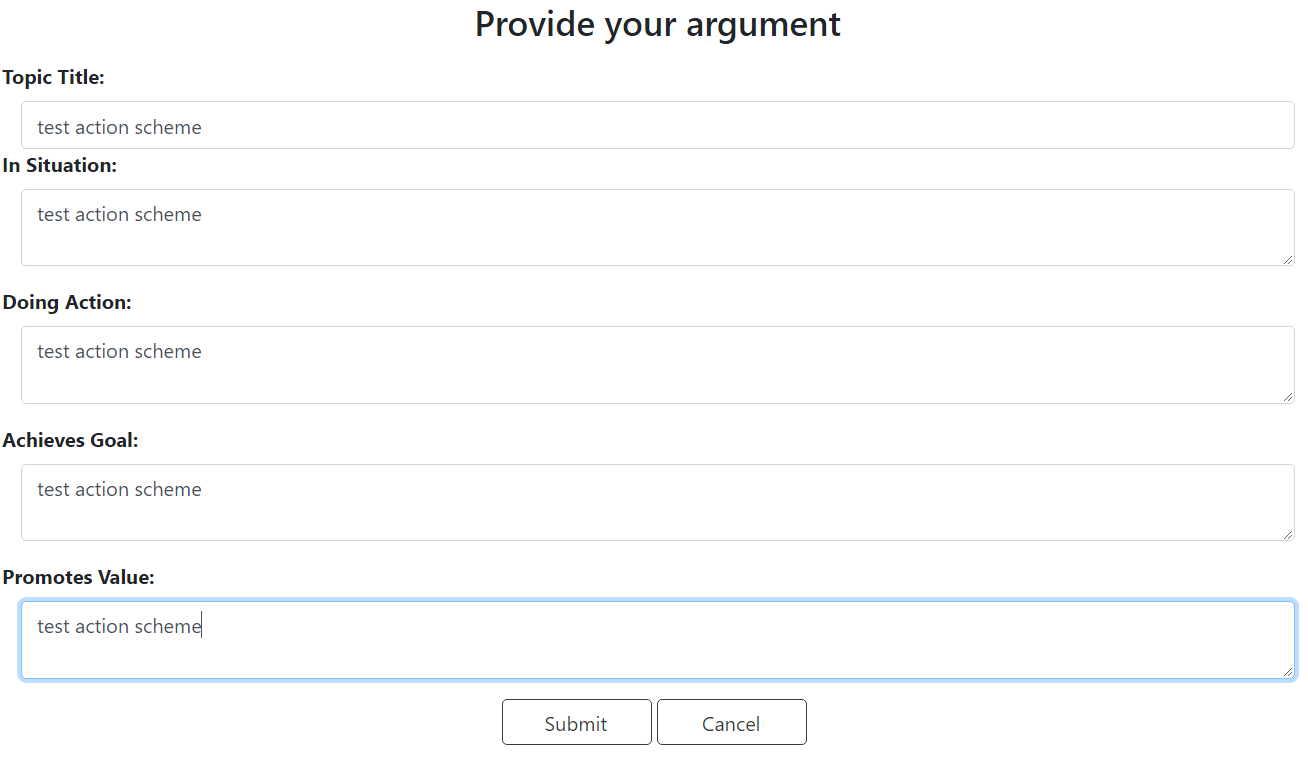
### A.3.4

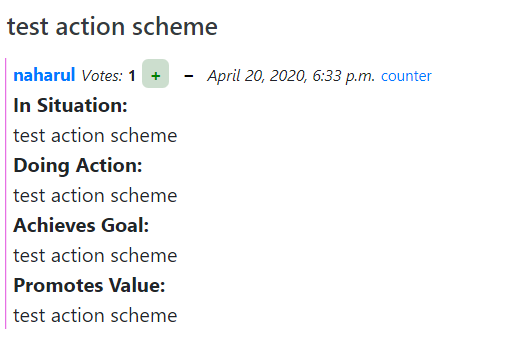


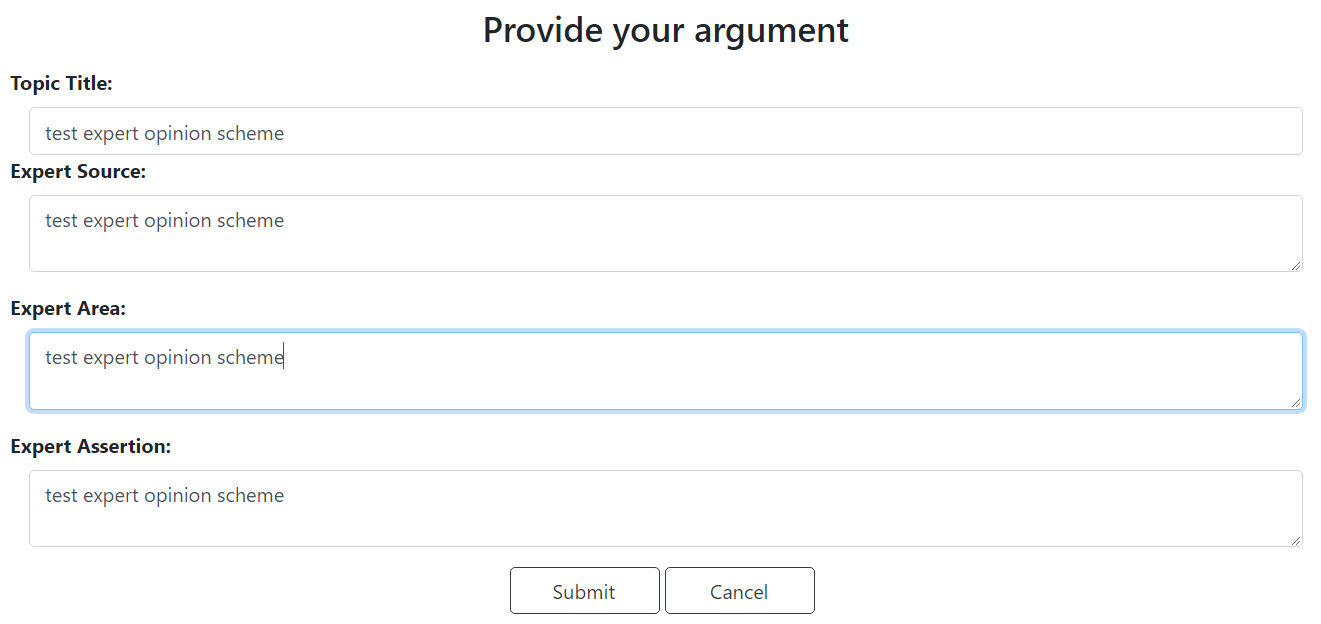
### A.3.5

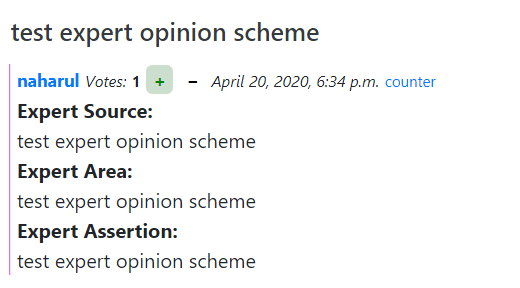
### A.3.6

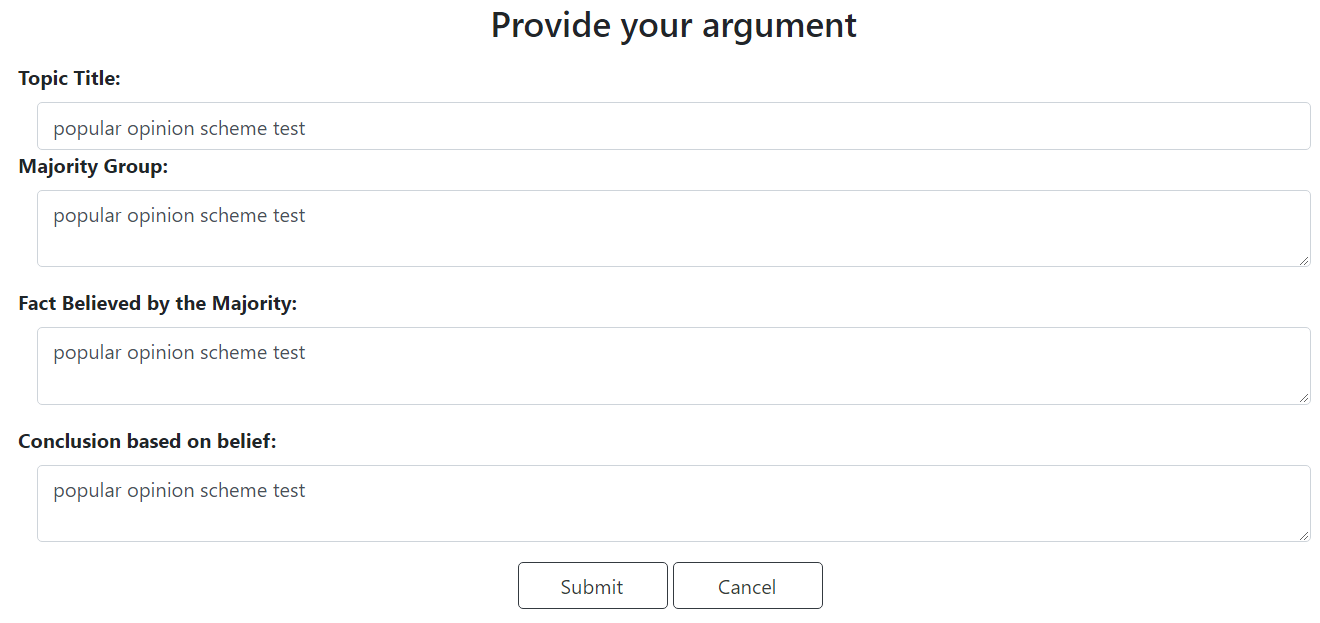


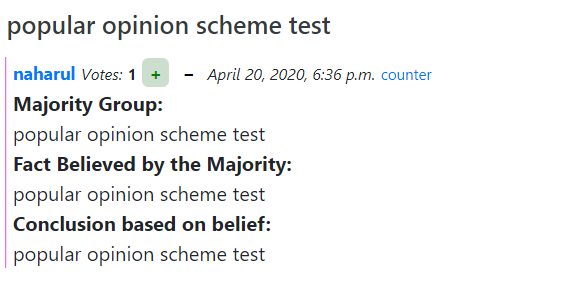


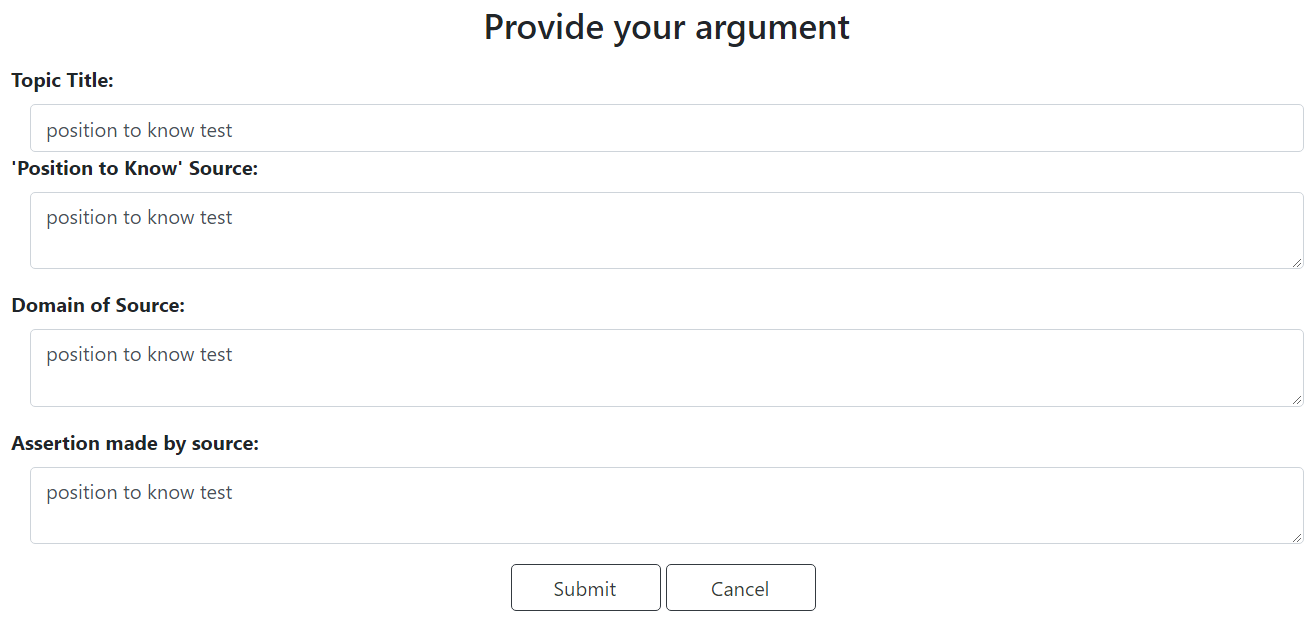


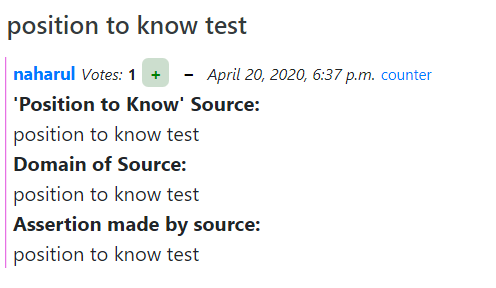


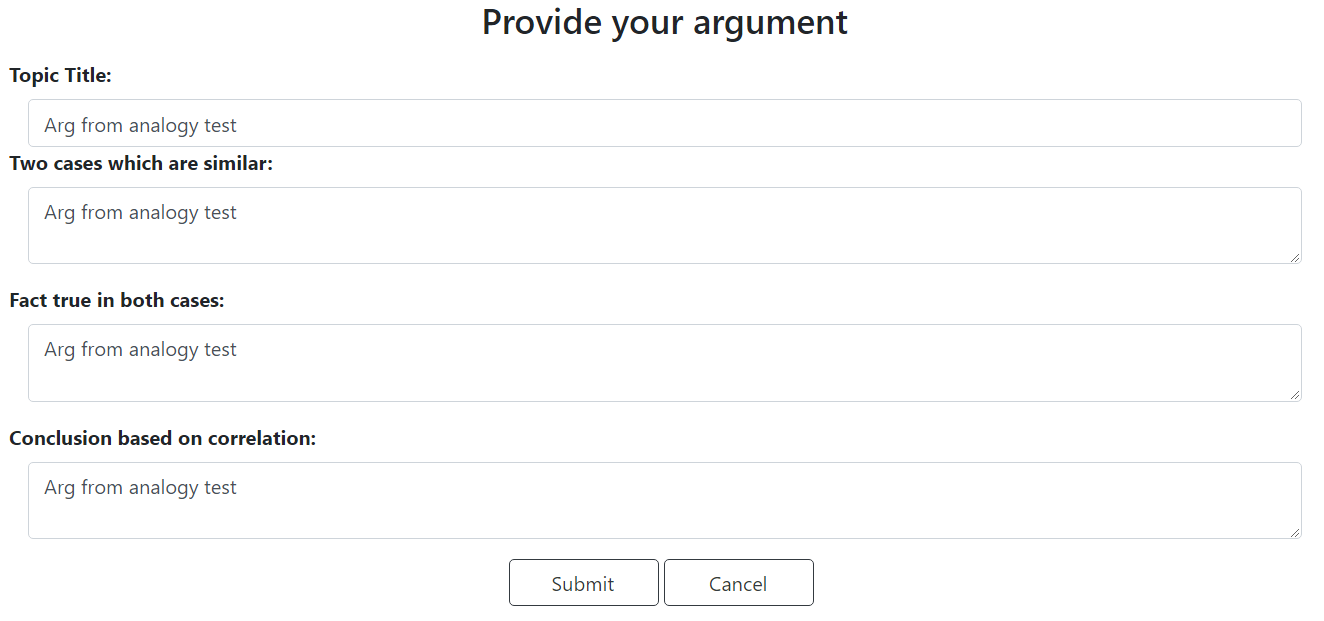


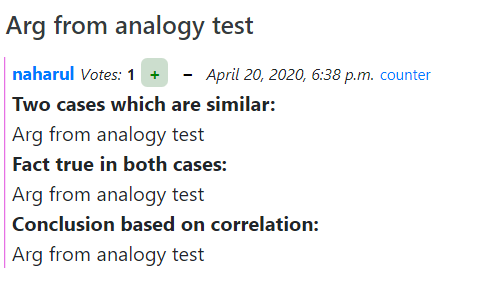


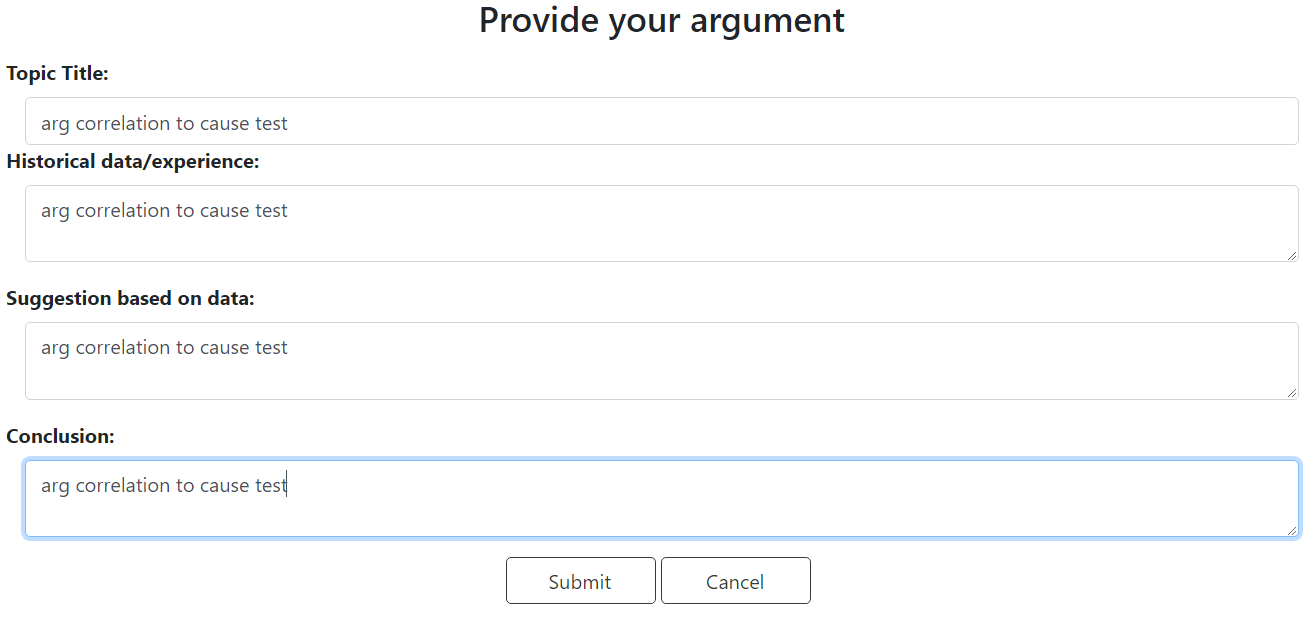


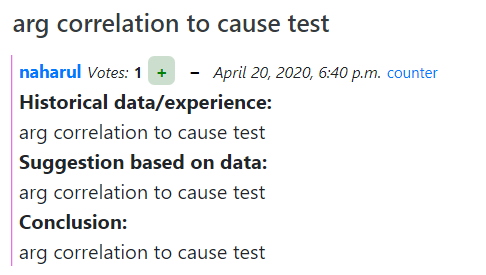




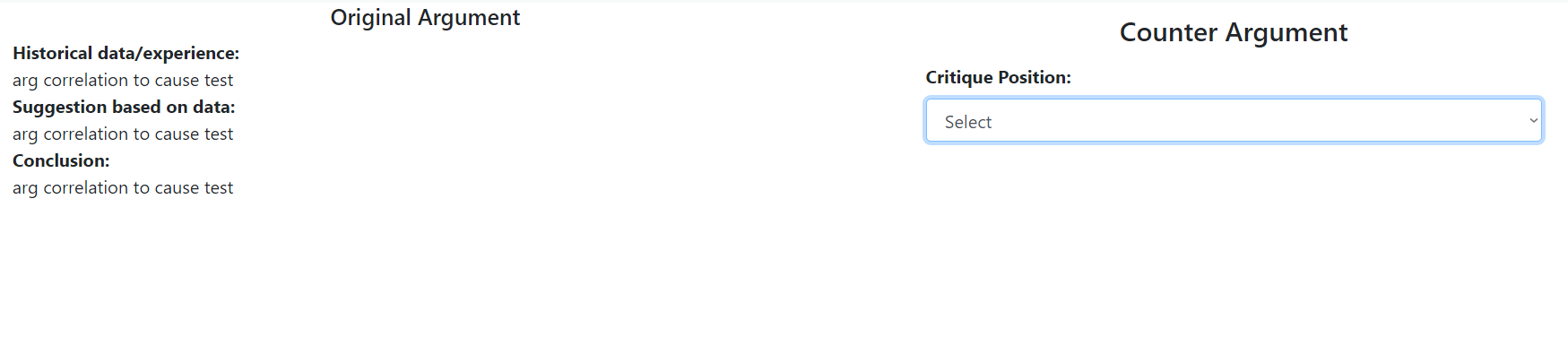


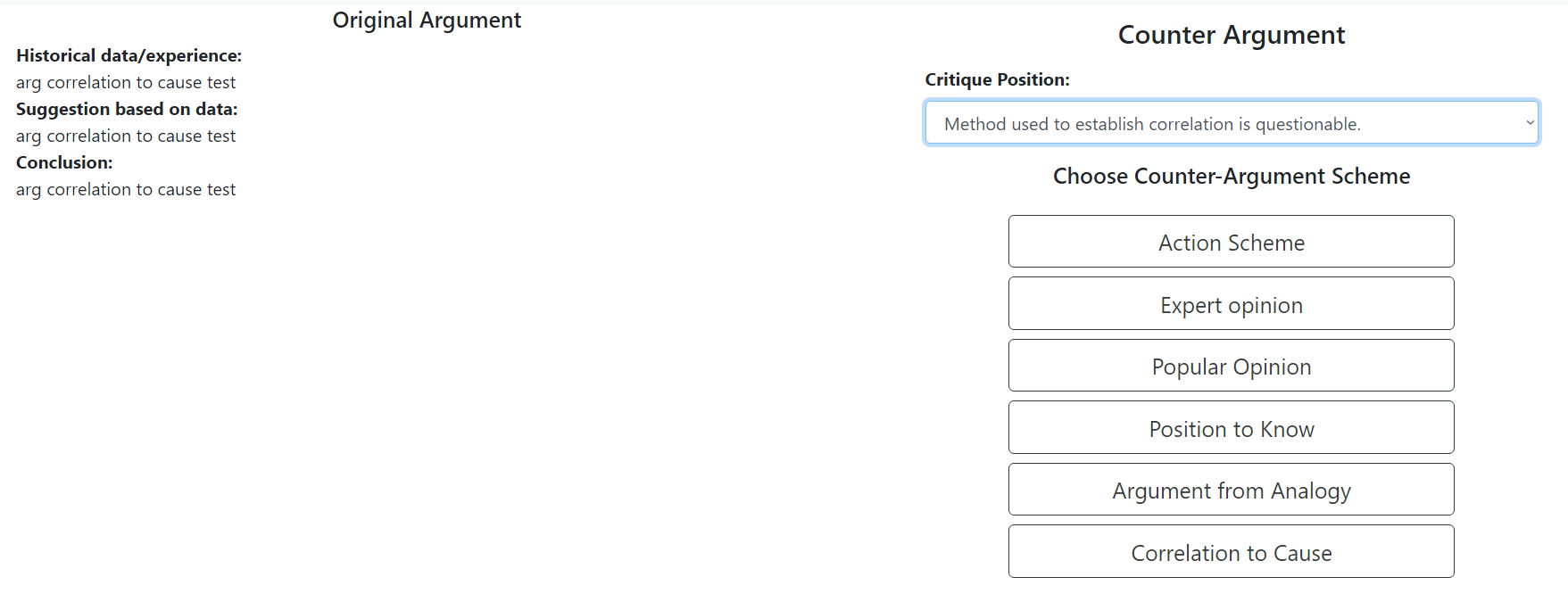


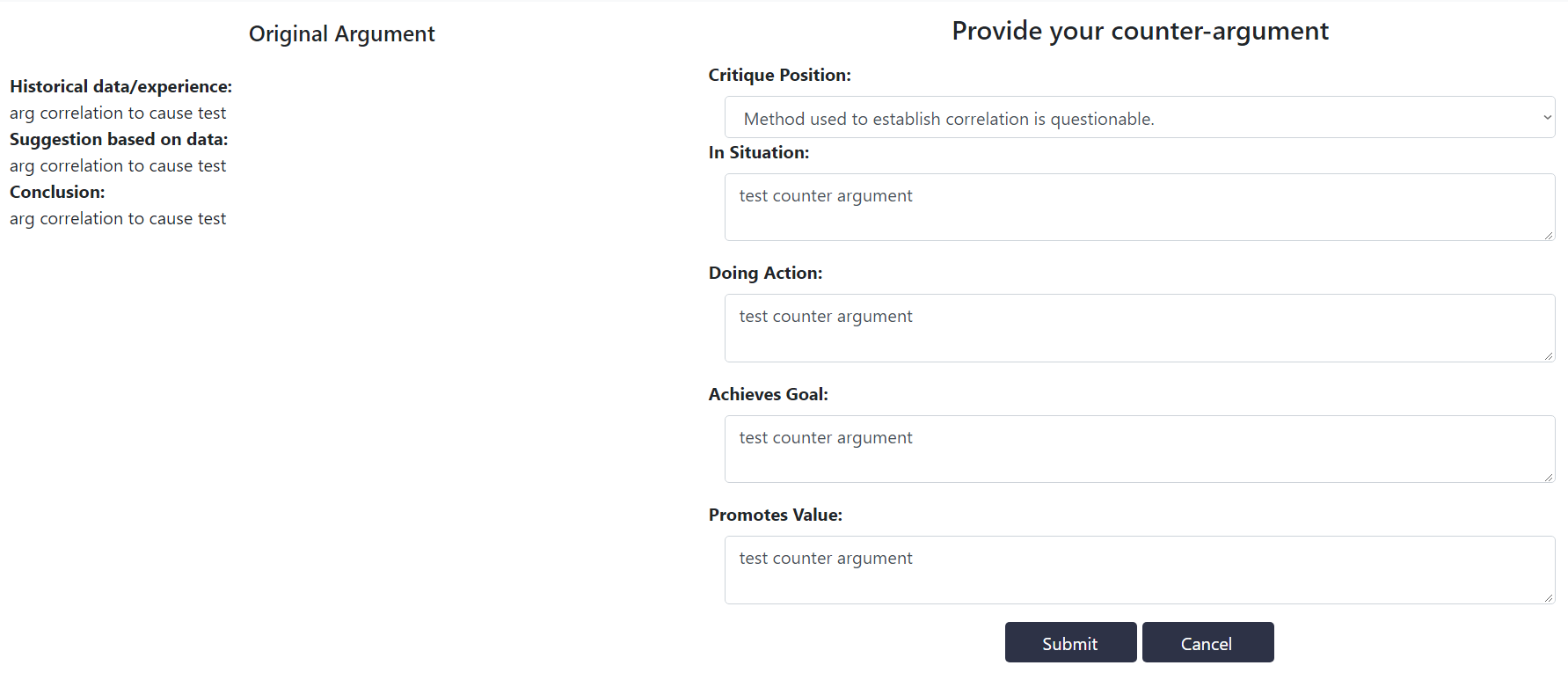


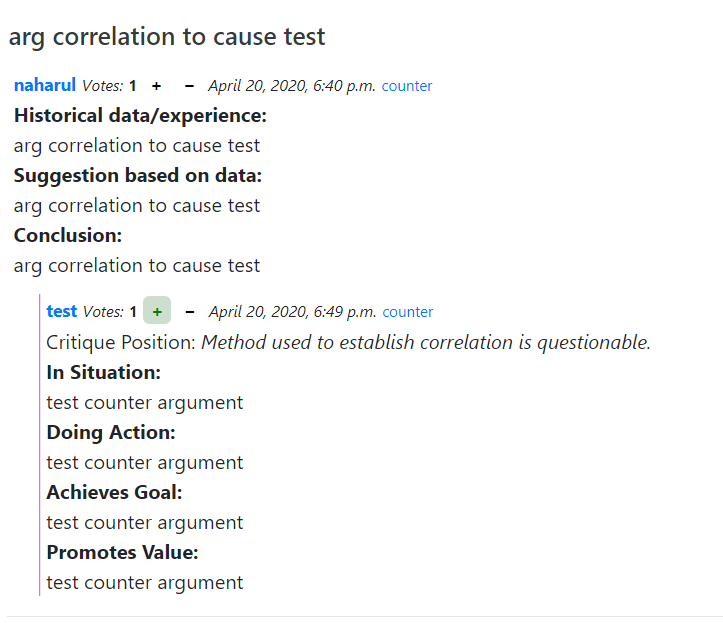


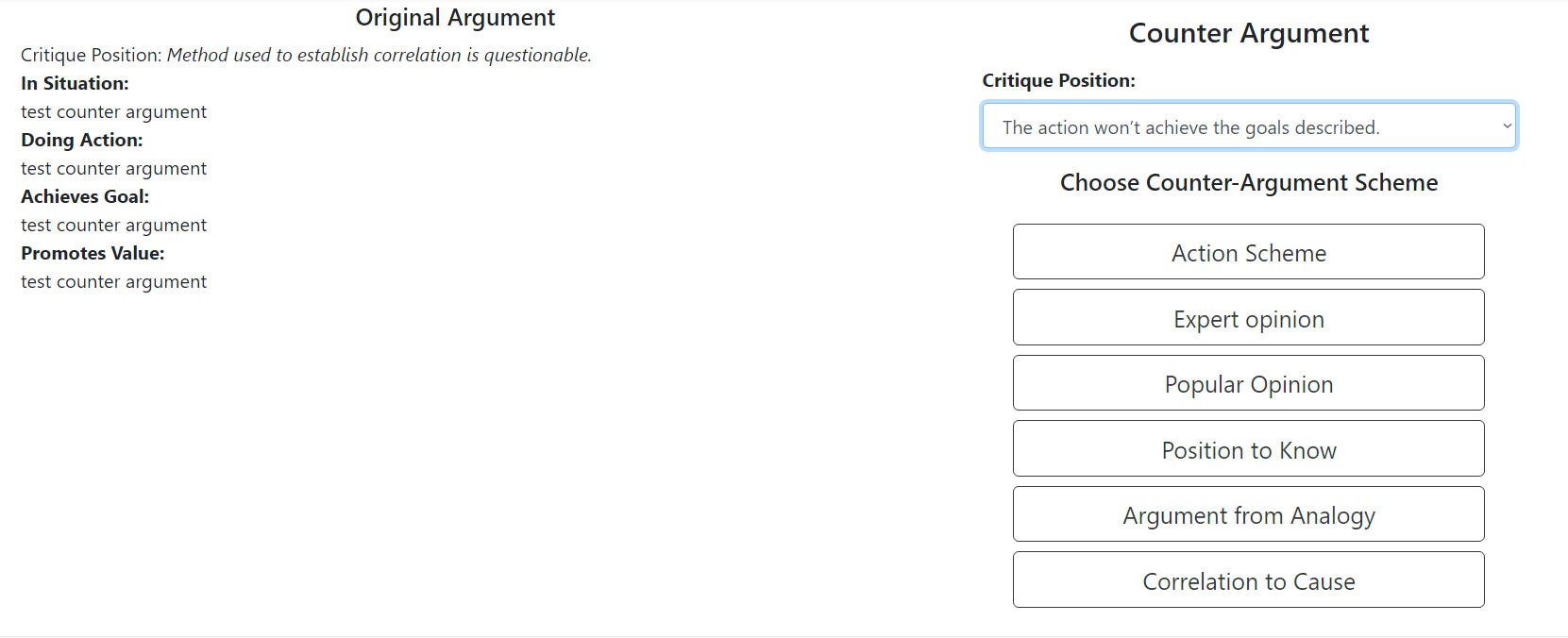
### A.3.7

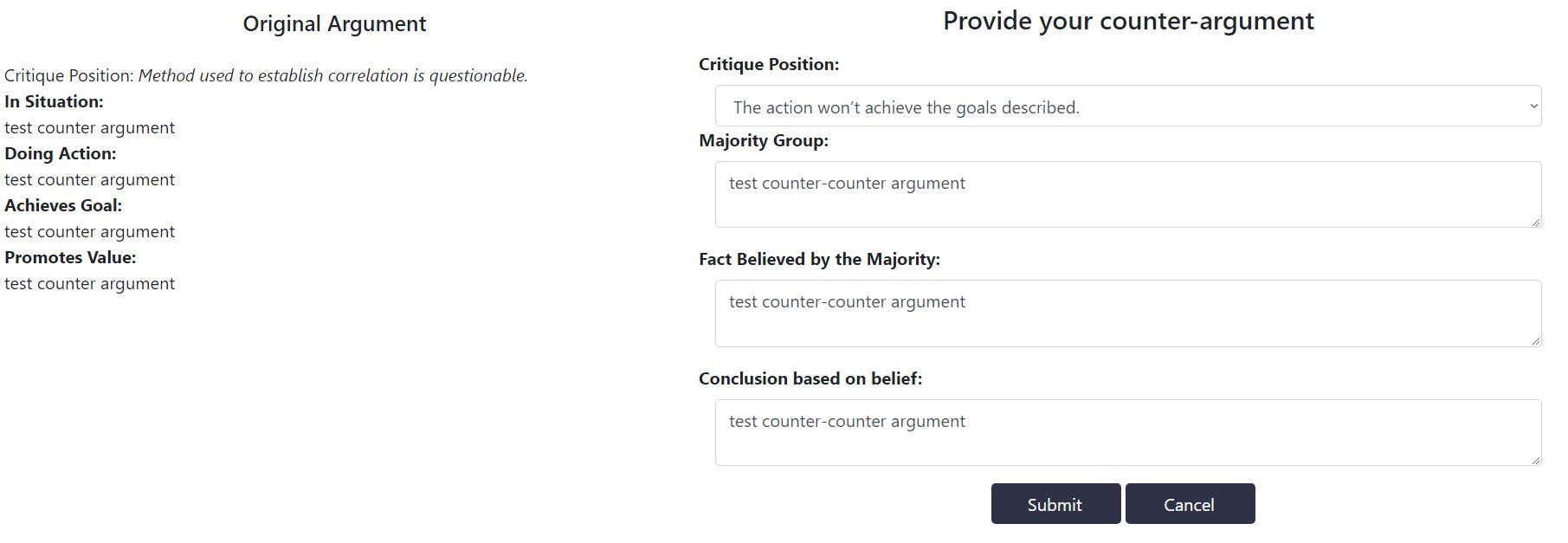


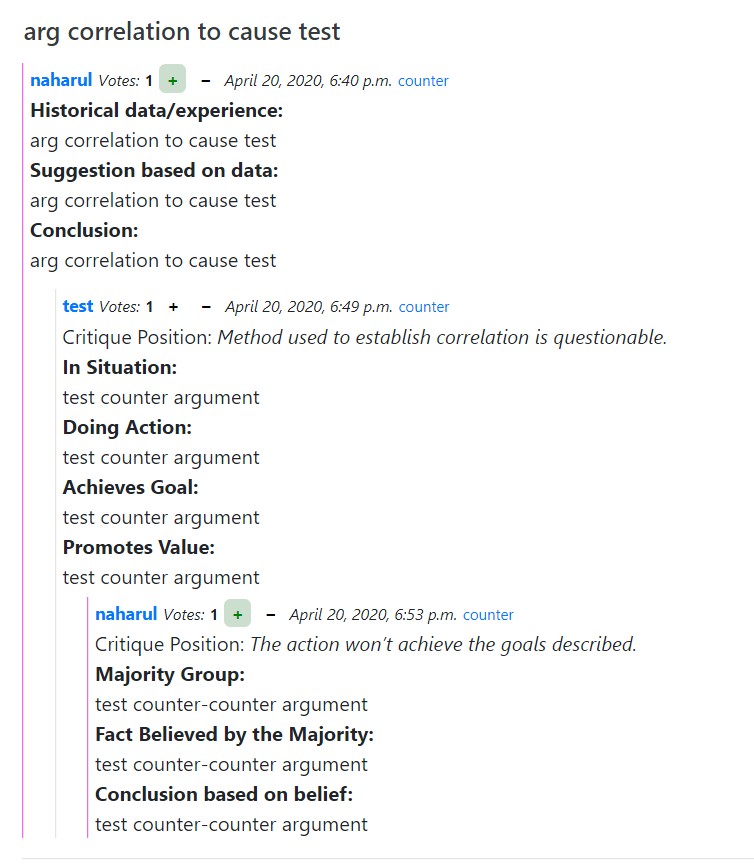




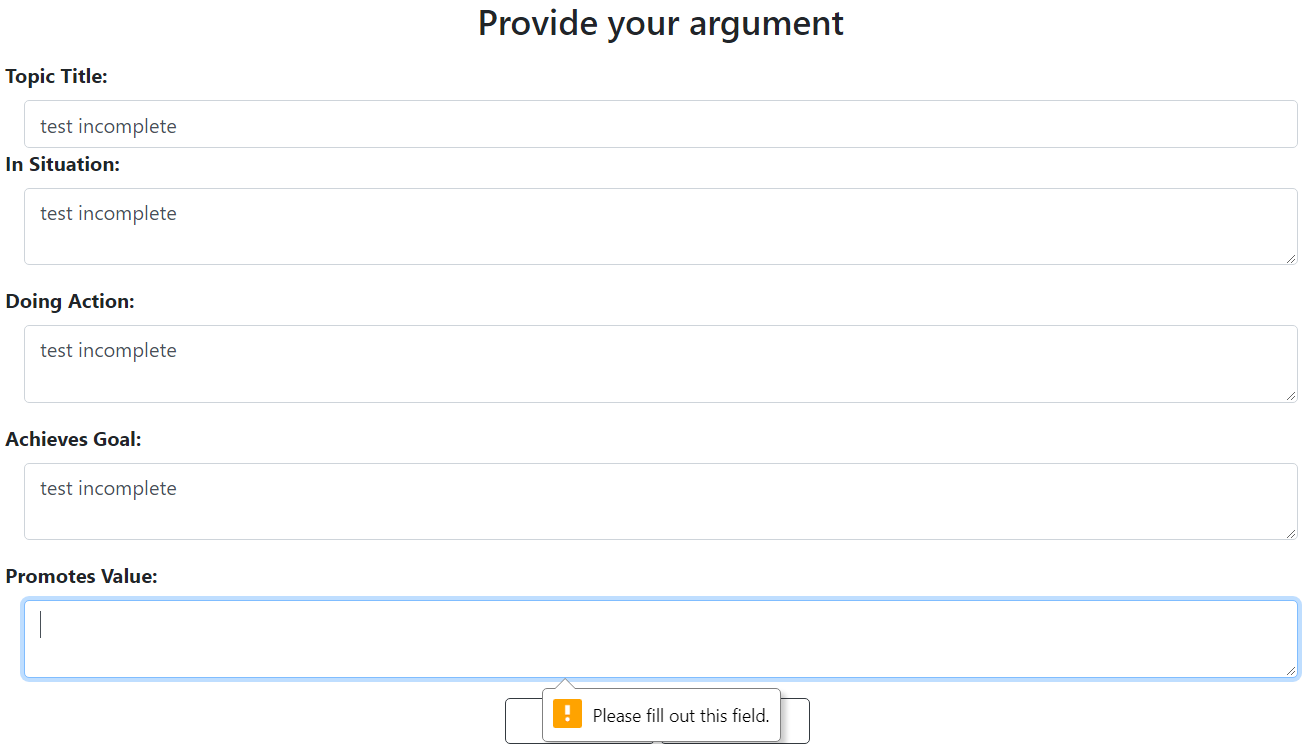




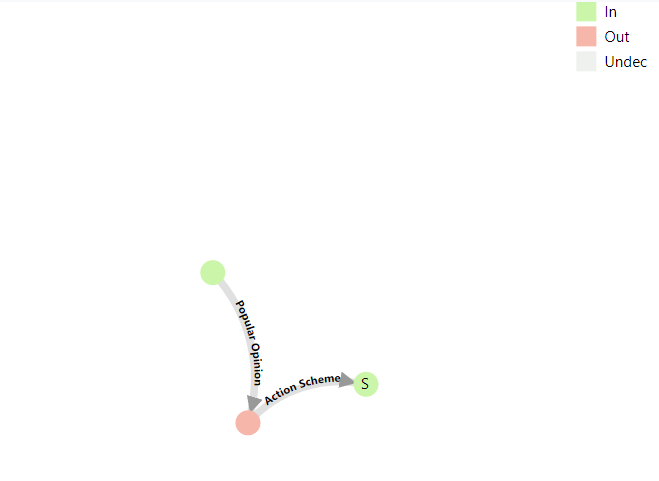




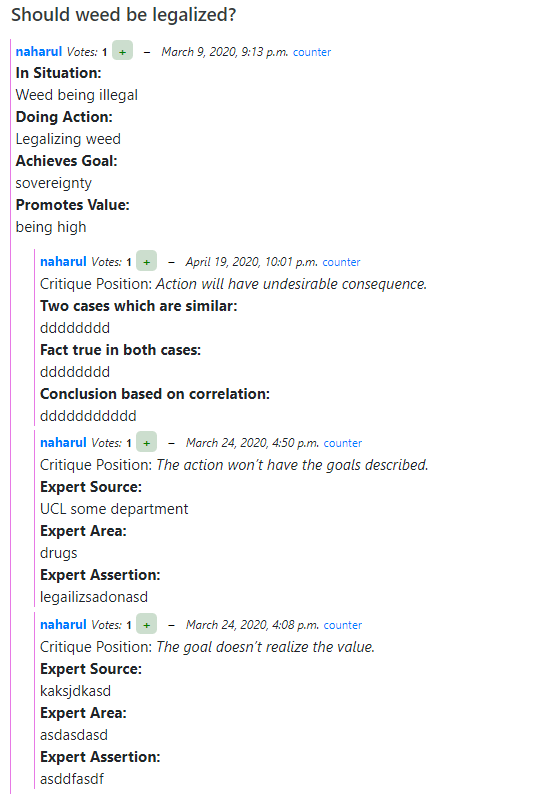
### A.3.8

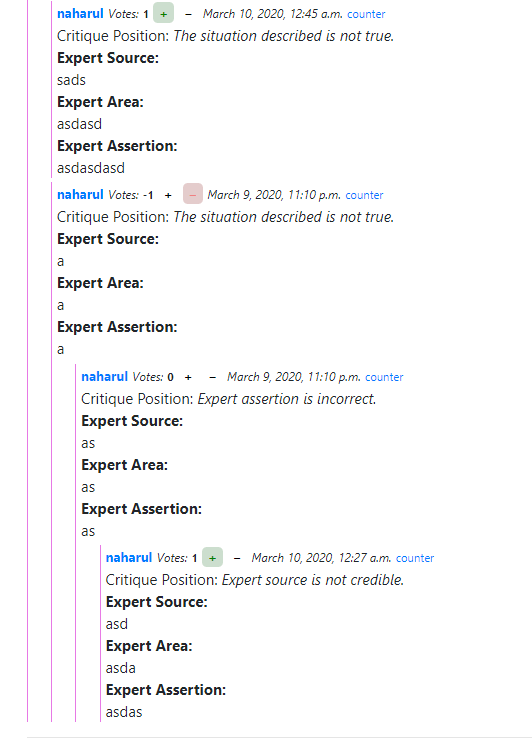


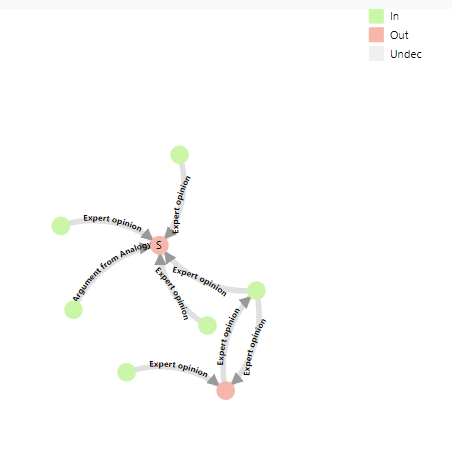
### A.3.9



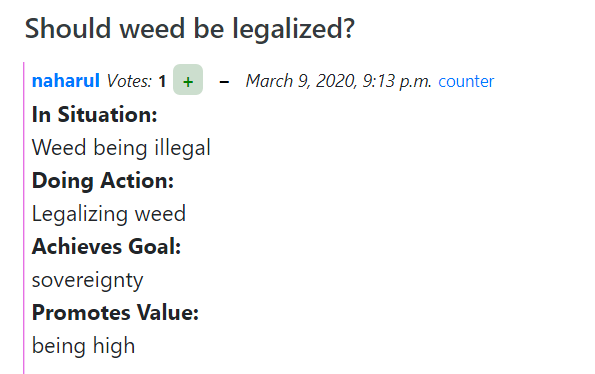
### A.3.10

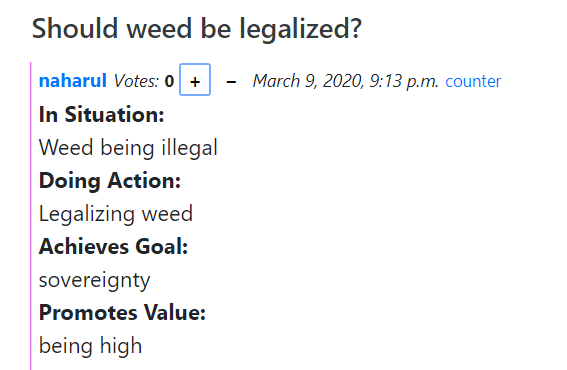


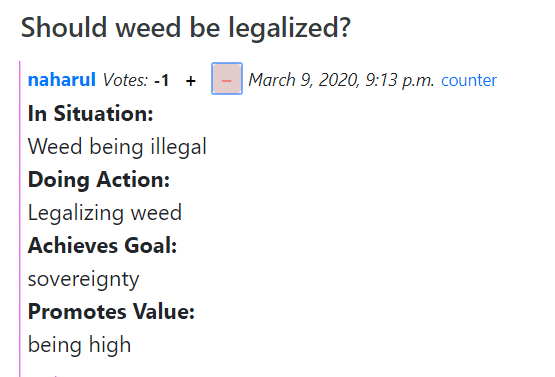




### A.3.11







# 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-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"**></**script**>  
 <**script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/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** : 300**px**; **padding-bottom** : 40**px"**>  
 <**div class="form-area"**>  
 <**form method="post"**>  
 {% **csrf\_token** %}  
 <**div class="container center\_div pl-3"**>  
 <**h3 style="margin-bottom**: 15**px**; **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**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**" role="button" value="Register"**/>  
 <**a href="**{% **url 'home'** %}**" class="btn btn-outline-dark" style="width**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**"**>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** : 300**px**; **padding-bottom** : 40**px"**>  
 <**div class="form-area"**>  
  
 <**form method="post"**>  
 {% **csrf\_token** %}  
 <**div class="container center\_div pl-3"**>  
 <**h3 style="margin-bottom**: 15**px**; **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**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**" role="button" value="Login"**/>  
 <**a href="**{% **url 'home'** %}**" class="btn btn-outline-dark" style="width**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**"**>Cancel</**a**>  
 </**form**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
{% **endblock** %}

### \templates\components\footer.html

<**footer class="section footer-classic context-dark bg-image" style="padding-bottom**:20**px**;**"**>  
 <**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"**)))  
 **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 substract 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**: 500**px"**>{{ **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**: 15**px**; **margin-top**: 10**px**;**text-align**: **center**;**"**>Counter Argument</**h4**>  
 <**label for="text" style="margin-left**: 15**px"**><**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**: 400**px"**>  
 {{ **scheme**.**scheme\_name** }}  
 </**a**>  
 </**div**>  
 {% **endfor** %}  
 </**div**>  
 </**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** : 40**px"**>  
 <**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** : 40**px"**>  
 <**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**: 15**px**; **margin-top**: 10**px**;**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**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**margin-left**:270**px**;**" role="button" aria-pressed="true" value="Submit"**/>  
 <**a href="**{% **url 'home'** %}**" class="btn btn-md active" style="background-color**:**#2d3246**; **color** :**white**; **width**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**"**>Cancel</**a**>  
 </**div**>  
 </**form**>  
 </**div**>  
 </**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** : 40**px"**>  
 <**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**: 15**px**; **margin-top**: 10**px**;**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**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**margin-left**:400**px**;**" role="button" aria-pressed="true" value="Submit"**/>  
 <**a href="**{% **url 'home'** %}**" class="btn btn-outline-dark" style="width**:120**px**; **padding-left**:10**px**;**padding-bottom**:5**px**;**"**>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**: 38**px**;**"**>  
 {% **if not entries** %}  
 <**h6**>Oops. There are no discussions to show.</**h6**>  
 {% **endif** %}  
 *<!-- mttp 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**: 1**px"**>  
 <**a href="**{% **url 'posts-detail-view' node**.**pk** %}**" style="margin-left**: 5**px**; **font-size**: 14**px**;**"**><**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***](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**: 38**px**;**"**>  
 {% **if not entries** %}  
 <**h6**>Oops. There are no discussions to show.</**h6**>  
 {% **endif** %}  
 *<!-- mttp 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**: 1**px"**>  
 <**a href="**{% **url 'posts-detail-view' node**.**pk** %}**" style="margin-left**: 5**px**; **font-size**: 14**px**;**"**><**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**: 12**px"**>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**: 15**px**; **font-size**: 12**px**; **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**: 12**px**;**"**></**div**>  
 {% **endif** %}  
 <**ul class="children entry" style="padding-left**: 25**px"**>  
 {{ **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) {  
 **$**(**`.downvote[data-id='**${data.**id**}**']`**).removeClass(**"user-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**: 650**px**; **width**: 1000**px**;**"**>  
 <**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'**)  
 .attrs({**'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**: 11**px !important**;  
 **padding**: 12**px**;  
 **background**: **rgba**(0, 0, 0, 0.8);  
 **color**: **#fff**;  
 **border-radius**: 2**px**;  
}  
.**d3-tip**:**after** {  
 **box-sizing**: **border-box**;  
 **display**: **inline**;  
 **font-size**: 10**px**;  
 **width**: 100%;  
 **line-height**: 1;  
 **color**: **rgba**(0, 0, 0, 0.8);  
 **content**: **"\25BC"**;  
 **position**: **absolute**;  
 **text-align**: **center**;  
}  
.**d3-tip**.**n**:**after** {  
 **margin**: -1**px** 0 0 0;  
 **top**: 100%;  
 **left**: 0;  
}  
  
  
**p** {  
 **display**: **block**;  
 **word-break**: **break-all**;  
 **word-wrap**: **break-word**;  
 **width**: 800**px**;  
  
}  
  
  
.**entry** {  
 **list-style**:**none**;  
}  
  
.**node** {  
 **font-size**: 12**px**;  
}  
  
.**node**.**positive** {  
 **color**: **green**;  
}  
  
.**node**.**negative** {  
 **color**: **red**;  
}  
  
.**node**.**user-upvoted** {  
 **color**: **white**;  
 **background-color**: **green**;  
  
 **padding**: 1**px**;  
 **border-radius**: 5**px**;  
 **border**: **none**;  
}  
  
.**node**.**user-downvoted** {  
 **color**: **white**;  
 **background-color**: **red**;  
  
 **padding**: 1**px**;  
 **border-radius**: 5**px**;  
 **border**: **none**;  
}  
  
.**downvote** {  
  
 **border-radius**: 5**px**;  
 **background-color**: **transparent**;  
 **font-size**: 14**px**;  
 **font-weight**: **bold**;  
 **border**: **none**;  
}  
  
.**upvote** {  
  
 **border-radius**: 5**px**;  
 **background-color**: **transparent**;  
 **font-size**: 14**px**;  
 **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**: -250**px**;  
 **visibility**: **hidden**;  
}  
.**entry-points** {  
 **font-size**: 12**px**;  
}  
  
.**usernode** {  
 */\* border-left: 1px solid rgba(8, 189, 147, 0.666); \*/* **border-left**: 1**px solid rgba**(218, 44, 215, 0.673);  
}  
  
.**usernode p**{  
 **margin-bottom**: 5**px**;  
}  
  
.**rootnode p**{  
 **margin-bottom**: 5**px**;  
}  
  
.**opnode** {  
 **border-left**: 1**px solid rgba**(15, 191, 255, 0.496);  
}  
.**commentnode** {  
 **border-left**: 1**px solid rgba**(0, 0, 0, 0.1);  
}  
  
*/\* Custom Markdown Styling \*/*.**entry-content** {  
 **margin-left**: 5**px**;  
 **word-wrap**: **break-word**;  
}  
.**entry-content h1**{  
 **font-size**: 18**px**;  
}  
  
.**entry-content h2**{  
 **font-size**: 16**px**;  
}  
  
.**entry-contentt h3**{  
 **font-size**: 16**px**;  
}  
  
.**entry-content h4**{  
 **font-size**: 16**px**;  
}  
  
.**entry-content h5**{  
 **font-size**: 16**px**;  
}  
  
.**entry-content h6**{  
 **font-size**: 16**px**;  
}  
  
.**entry-content pre code** {  
 **background-color**: **rgb**(246, 246, 246);  
}