**Blogging Application**

****

**Readme Document**

**Author: Ghazala Khanam**

**CONTENTS**

INTRODUCTION

............................................................................................................................................. 3

REQUIREMENTS

............................................................................................................................................. 3

API LEVEL DESCRIPTION

............................................................................................................................................. 3-4

API DATA MODEL (Resources):

............................................................................................................................................. 4-5

DATABASE PACKAGES:

............................................................................................................................................. 5-6

CACHING- MEMCACHED

............................................................................................................................................. 6-7

INSTALLATION

............................................................................................................................................. 7

SCREENSHOTS

............................................................................................................................................. 7-13

**INTRODUCTION**

---------------------------------------------------------------------------------------------------------------------

Blogging application is an web based application where end users can access through web browser and logging to the application. This application is developed using django 1.11.13 (commands, template tags, statics, media files, etc).

* UI to view list of article posted on application.
* All users have access to view posted articles and its comments.
* User ( only registered user) can add new article with image content.
* Registered Users can also post comments on the article .

**REQUIREMENTS**

---------------------------------------------------------------------------------------------------------------------

Django 1.11.\*

Python >= 2.7

python-memcached

**API LEVEL DESCRIPTION**

---------------------------------------------------------------------------------------------------------------------

**This Application have following concepts:**

**Blog**: The root concept of the API, it has posts and pages. This is the container for blog meta-information like Blog Name and Description.

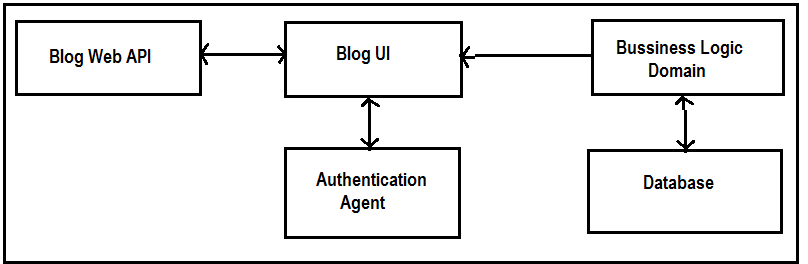
**Articles**: A publishable item that the blog author has created.

**Comments**: A place where user can reacts to what the author has written.

**Pages**: A page is a place for static content, such as biographical information, or ways to contact the user.

**Users**: Someone who interacts with Blogging application, an Author, an Administrator, or a Reader.

**Major components:**



1. Blog UI : This is user interface layer developed using HTML, Bootstrap.
2. Blog Web API : This is a REST service build on Django framework version 1.11.13 responsible to provide data to User Interface
3. Authentication: This component validate user to login to Blogging application.
4. Business Logic Domain: All business logic of Application is encapsulated in this component.
5. Database: SQLite Database is used for the project.

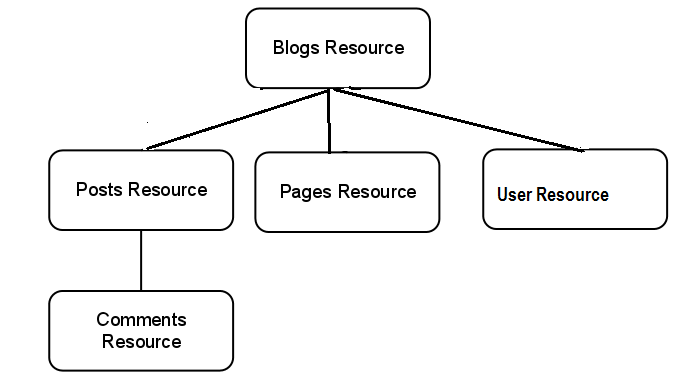
Django ENGINE value : django.db.backends.sqlite3

**Are external physical component connections needed?**

**No,** The portal connects directly to the application server and does not require any external components.

**API DATA MODEL (Resources):**

---------------------------------------------------------------------------------------------------------------------

****

**Blogs Resource:** Represents a blog.

**Articles Resource:** Represents a post; each Posts Resource is a child of a Blogs Resource.

**Comments Resource**: Represents a comment on a specific article, each Comments Resource is a child of a Articles Resource.

**Pages Resource:** Represents a static page; each Pages Resource is a child of a Blogs Resource.

**Users Resource:** Represents a non-anonymous user.

**DATABASE PACKAGES:**

---------------------------------------------------------------------------------------------------------------------

**Database Model:**

In this project we are using SQLite DB with Django, which encapsulates databases tables through models. Essentially, a model is a Python object that describes data model/table. In This Blogging Application we have 3 models/tables as below:

**Article:**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Key** |
| Id (Default) | AutoField | Primary Key |
| Title | Varchar (60 char) |  |
| Slug | SlugField |  |
| Body | TextField (140 char) |  |
| Date | DateTimeField |  |
| Thumb | ImageField |  |
| Author | User Model | Foreign Key |

**Comment:**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Key** |
| id (Default) | AutoField | Primary Key |
| Post | Article | ForeignKey |
| Author | User Model | ForeignKey |
| Body | TextField |  |
| created\_date | DateTimeField |  |
| approved\_comment | BooleanField |  |

**User:** User model is Django's Built-in model for Auth system available in django.contrib.auth engine, some of the fields of user model are as below:

|  |  |
| --- | --- |
| **Field** | **Type** |
| Username | CharField |
| Password | password |
| Email | EmailField |
| first\_name | CharField |
| last\_name | CharField |

**How to migrate operation to database tables:**

$ cd projectname/

$ python manage.py migrate

**CACHING-** **MEMCACHED**

---------------------------------------------------------------------------------------------------------------------

In this project we have used MEMCACHED caching mechanism.

**Memcached:**

Memcached is an open source , memory-based caching system that support Python bindings through python-memcached and can run as a daemon.

Memcached is running on localhost (127.0.0.1) port 11211, using the python-memcached binding:

**key** : Title of Article.

**Value**: Data of cached article.

**Cache** **Time**: 15 mins (900 secs)

**INSTALLATION**

---------------------------------------------------------------------------------------------------------------------

To setup a local development environment we must run below commands on commands prompt:

• $ cd projectpath/

• $ python manage.py migrate

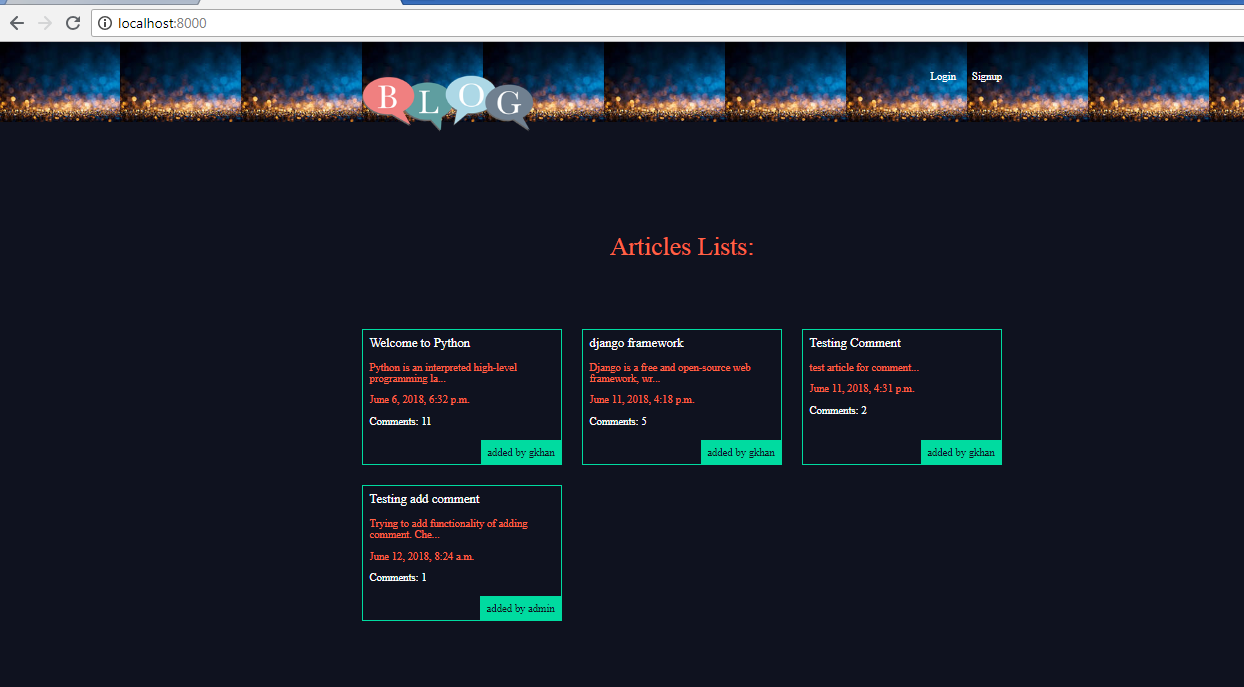
• $ python manage.py runserver

**Starting Web Browser:** $ python manage.py runserver

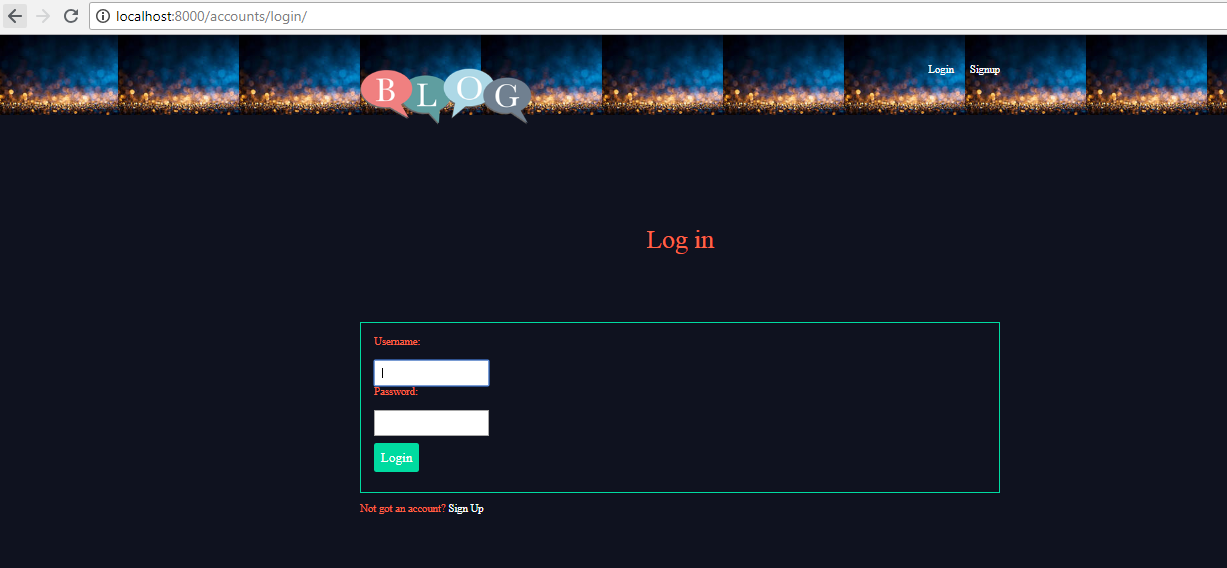
**SCREENSHOTS**

---------------------------------------------------------------------------------------------------------------------

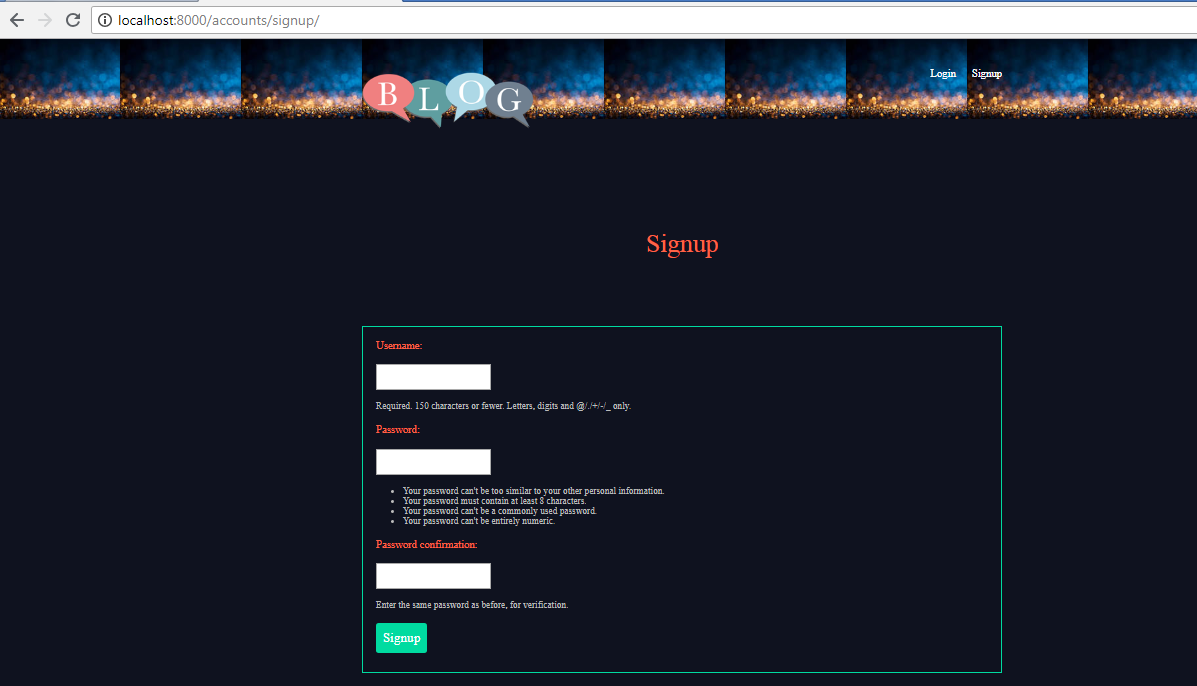
**Browse URL**: http://127.0.0.1:8000/



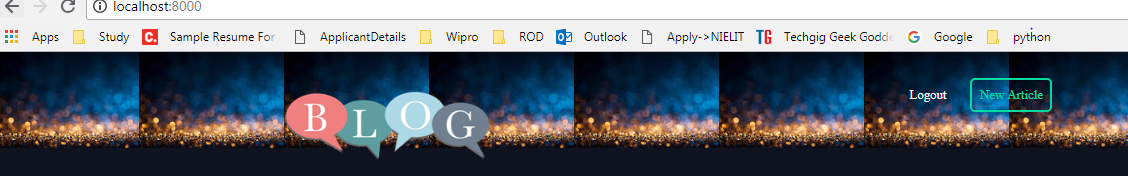
**Login**: Registered user can login to the blog. URL: http://localhost:8000/accounts/login/



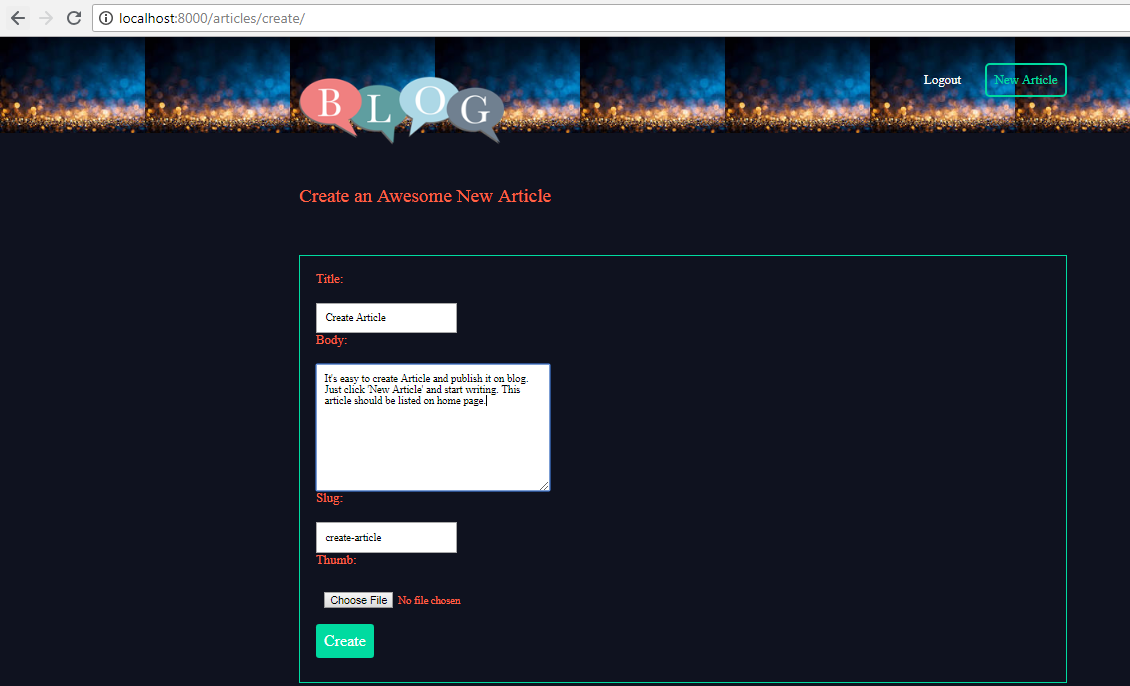
**Signup**: New user can register using signup. URL: http://localhost:8000/accounts/signup/



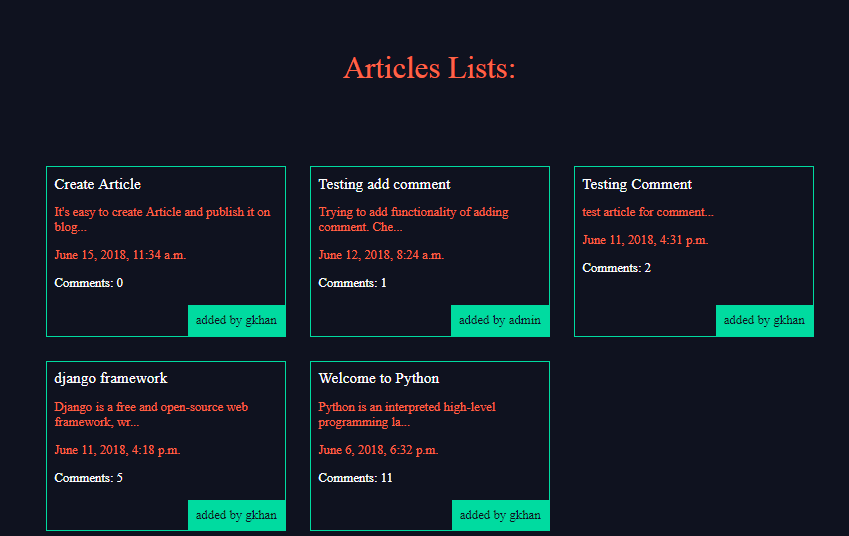
**New Article**: To add new Article to blog.



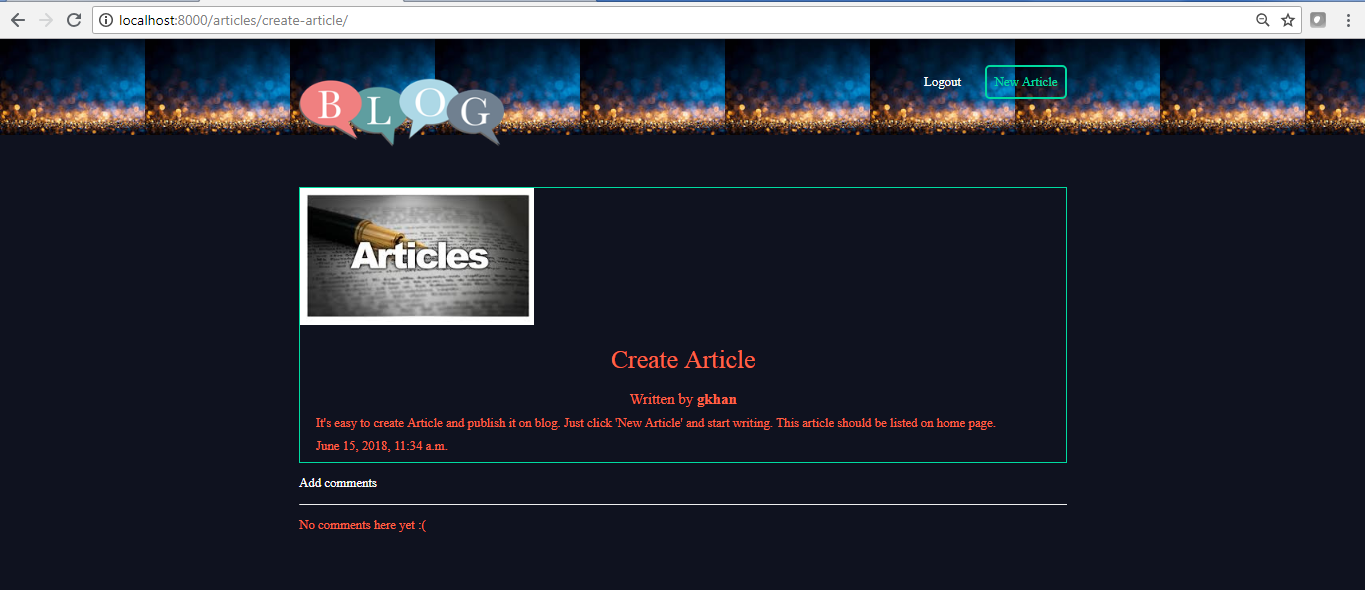
Create Article:



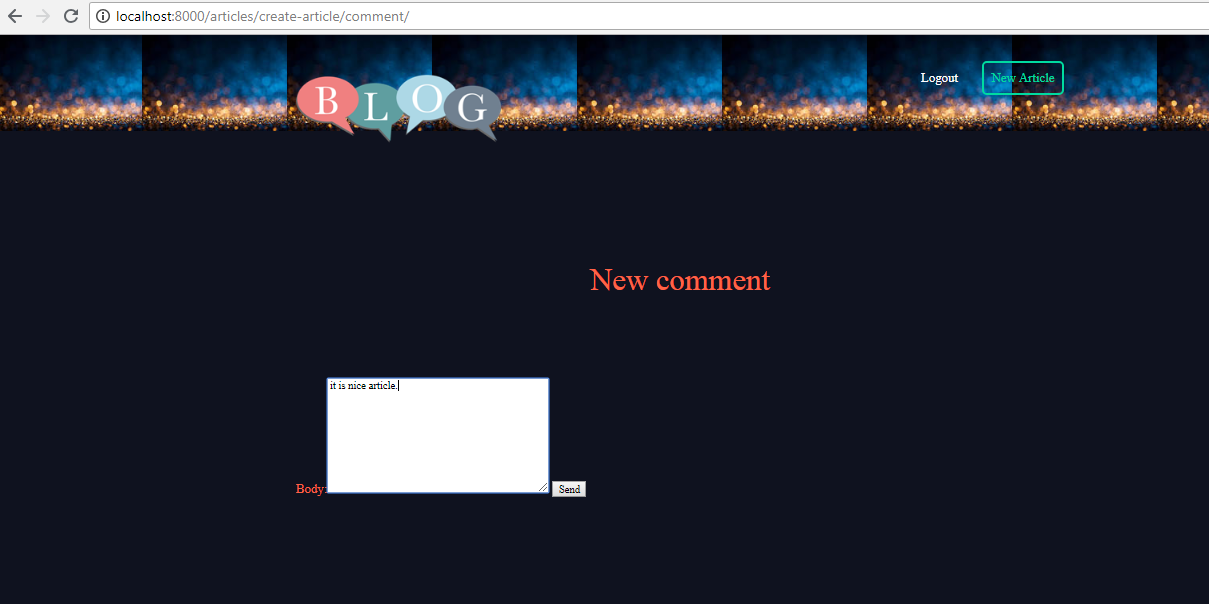
Article List:



**Article\_Details**: To view complete article details. URL http://localhost:8000/articles/create-article/

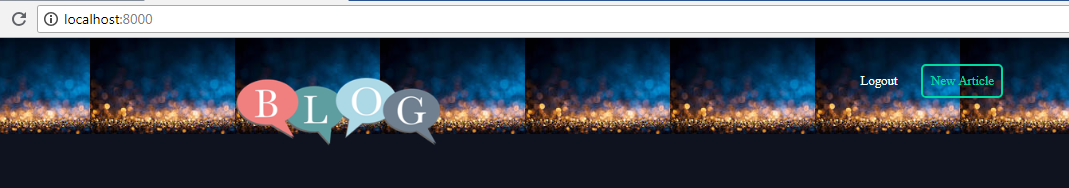


**Add\_Comment:** This module is used to add comments to the article.





**Logout**: logout from the blog.



**Note**:

* All articles will be listed in descending order by date.
* Only authorized users can write article of the blog.
* Only authorized users can comment on the article.