In The Name of Allah

This project is for scienceofcoding.ir website. Before that, I used Ali\_Blog\_Django\_1\_BookShop project. But in 16th Azar 1396, I want to change my site completely, remove some part and add another part.

**Purposes:**

* Writing articles about programming, network, linux and ...
* Writing comments for articles
* Introduce small project about python and upload these projects in GitHub.
* Wedsite analysis like piwik and google analytic
* Display online users
* Get IP of users when open this website
* Multi language
* In frondend : Angular 4
* In backend : Django REST

**Article**

In this part I want to write some articles about programming in python, django, angular and so on. Also I want to write articles about network, security and linux.

For this part I need these models:

* Article
* Category
* Tag

Article have two part:

* Article
* Small project

Category model:

* Name: charField(length=50)
* Slug: slugField()
* Image: imageField

Property for category model:

* Get\_absolute\_api\_url: return url of category
* \_\_str\_\_: string of class
* Upload location image: for image of category
* Create slug: for creating slug for category. Slug should be unique
* Pre\_save\_category\_receiver: a method for doing sth before save category obj.
* Get\_articles: get articles related to this category.

Tag Model:

* Name: charField(50)
* Slug: SlugField()

Properties for Tag model:

* Get\_absolute\_api\_url: return url of tag
* \_\_str\_\_: string of class
* Create slug: for creating slug for tag. Slug should be unique
* Pre\_save\_tag\_receiver: a method for doing sth before save category obj.
* Get\_articles: get articles related to this tag.

Article Model:

* Title: CharField(length=120)
* Content: TextField or richTextField from ckeditor or markdown or pagedown
* Author: foreign key with user
* Image: ImageField, for upload locatin use Upload\_location\_image() method
* Slug: SlugField and related to title.
* View\_count: how many times users view this article.
* Is\_active: if True, display article
* Featured: for home page
* Category: foreign key with Category model
* Tags: many to many with Tag model
* Created\_datetime: datetime article created
* Updated\_datetime: datetime article updated
* Objects = ArticleManager()

Propery for article model:

* Class ArticleManager:
  + Function active\_all: return all function that are active.
* Get\_absolute\_api\_url: return url related to the article
* Create slug: create slug for article before save on db
* Pre\_save\_article\_receiver: a method for doing sth before save article obj.
* Get\_comments: get comments related to this article.
* Upload\_location\_image: location of image and create a name for image related to article title
* ordering = ['-created\_datetime’, '-updated\_datetime']

**Users:**

* Create a session

**Comment**

Client user can write comment for each article.

For comment, I have two ideas:

* I use Linkedin authentication
* Write a comment app, that take username and email from client user.

**Website Analysis**

In this part, I want to present a system like piwik and google analytic.

* Get ip of user, who open my website.
* Get email
* Get image of him or her
* Take some information from his browser
* Is online or not
* Display online users
* Which page, he is watching now
* When he have started watching page and how much
* For each page, how many times viewed by the user

For this job, I need to learn something:

* How to get client user’s IP: test getting ip in scienceofcoding.ir, have a test page and do my testing on it. Also test *django-ipware* and verify which of them is better.
* Find user location by Geolocation.
* Display online users: I think *django channel* is good idea for this. Also verify ohter ways for this job.

**Admin**

Sections in Admin part:

**Articles:**

* Display articles
* Counts of comment for each article

**Comments:**

* Display parent comments

**Users:**

* Display online users
* Display each online users, looking which page

**Diagram:**

* For displaying statistics, I want to use disgrams and charts