Django Workshop

Day 2

Django Models

A **model** is basically a **class** that represents a **table or collection** in our **Database**. Which contains all the information regarding the table. These models are stored together in Django in a file **models.py** in our App.

SQL

```sql
CREATE TABLE article (
    id NOT NULL PRIMARY KEY,
    title varchar(255) NOT NULL,
    body text NOT NULL,
    publish_date date NOT NULL
);
```

Python ORM

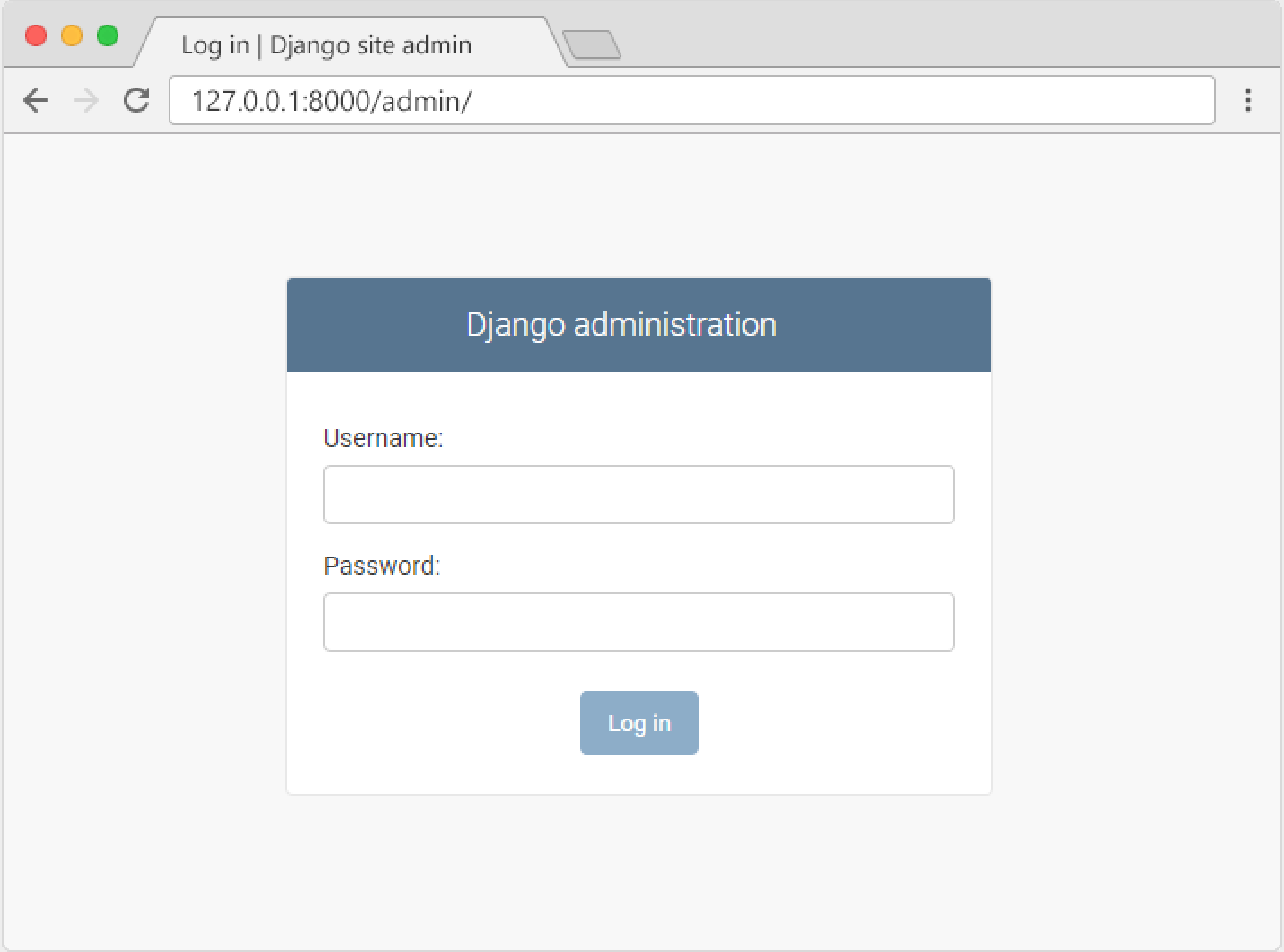
```
class Article(models.Model):
    title = models.CharField(max_length=255)
    body = models.TextField()
    publish_date = models.DateField()
```

```
python manage.py makemigrations

python manage.py migrate
```

Django Admin Site

The Django admin site is a built-in interface that allows developers to manage and view the application's data through a user-friendly web interface. It provides tools for adding, editing, and deleting records in the database.



Log in | Django site admin

127.0.0.1:8000/admin/

Django administration

Username:

Password:

```
python manage.py createsuperuser
```

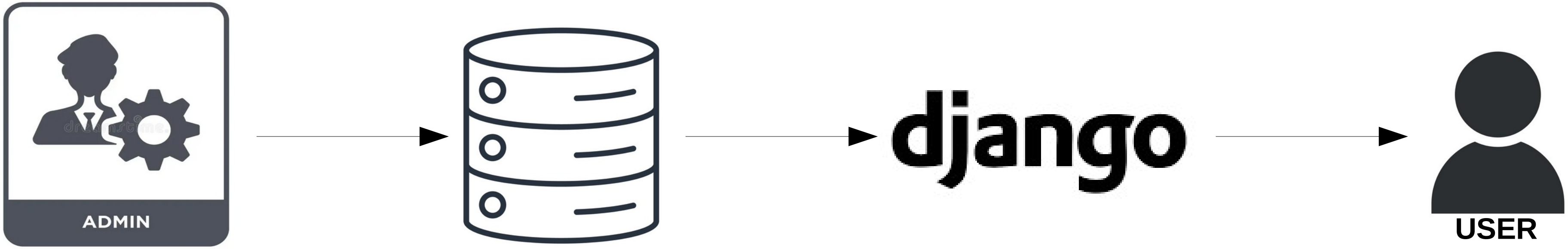
SQL

```
SELECT * FROM article;
```

Python

```
Article.objects.all()
```

DATA RECORDS



SQL

```
SELECT * FROM article
ORDER BY publish_date DESC;
```

Python

```
Article.objects.all().order_by("-publish_date")
```

SQL

```
SELECT * FROM article
WHERE id = article_id;
```

Python

```
Article.objects.get(id=article_id)
```

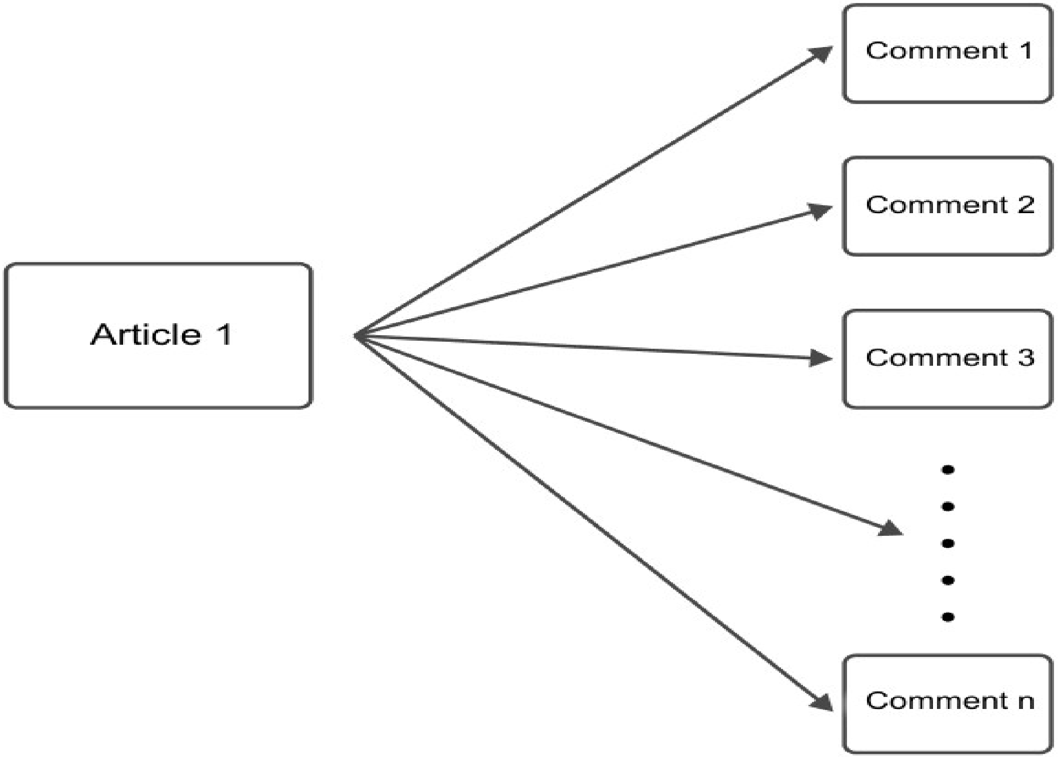
SQL

```
CREATE TABLE article (
    id NOT NULL PRIMARY KEY,
    title varchar(255) NOT NULL,
    body text NOT NULL,
    publish_date date NOT NULL
    image_link text NOT NULL;
);
```

SQL

```
CREATE TABLE comment (
    id serial NOT NULL PRIMARY KEY,
    name varchar(100) NOT NULL,
    body varchar(2000) NOT NULL,
    article_id integer NOT NULL REFERENCES article(id)
);
```

Many to One Relationship



Python

```python
class Comment(models.Model):
    name = models.CharField(max_length=100)
    body = models.CharField(max_length=2000)
    article = models.ForeignKey(
        Article,
        on_delete=models.CASCADE,
    )
```

SQL

```
SELECT * FROM comment
WHERE article_id = 1;
```

Python

```
Comment.objects.filter(article=article)
```

SQL

```
INSERT INTO comment (name, body, article_id)
VALUES ('Tejaswin', 'Great article!', 2);
```

Python

```
Comment.objects.create(name=‘Tejaswin’, body=‘Great article!’,
article_id=2)
```