INTERNSHIP AT CREART SOLUTIONS PVT. LTD.

AN INTERNSHIP REPORT

ON

NOTICEHUB USING D’JANGO PYTHON

Submitted by

Parmar Mihir Mehulbhai

(200390116051)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

Information Technology Department

S.P.B. Patel Engineering College, Mehsana

Gujarat Technological University, Ahmedabad

July-August, 2023

S.P.B. PATEL
ENGINEERING COLLEGE
SAFFRONY INSTITUTE OF TECHNOLOGY

GUJARAT TECHNOLOGICAL UNIVERSITY

S.P.B. Patel Engineering College

Near Shanku's Water Park, Ahmedabad – Mehsana Highway, Linch, Gujarat

CERTIFICATE

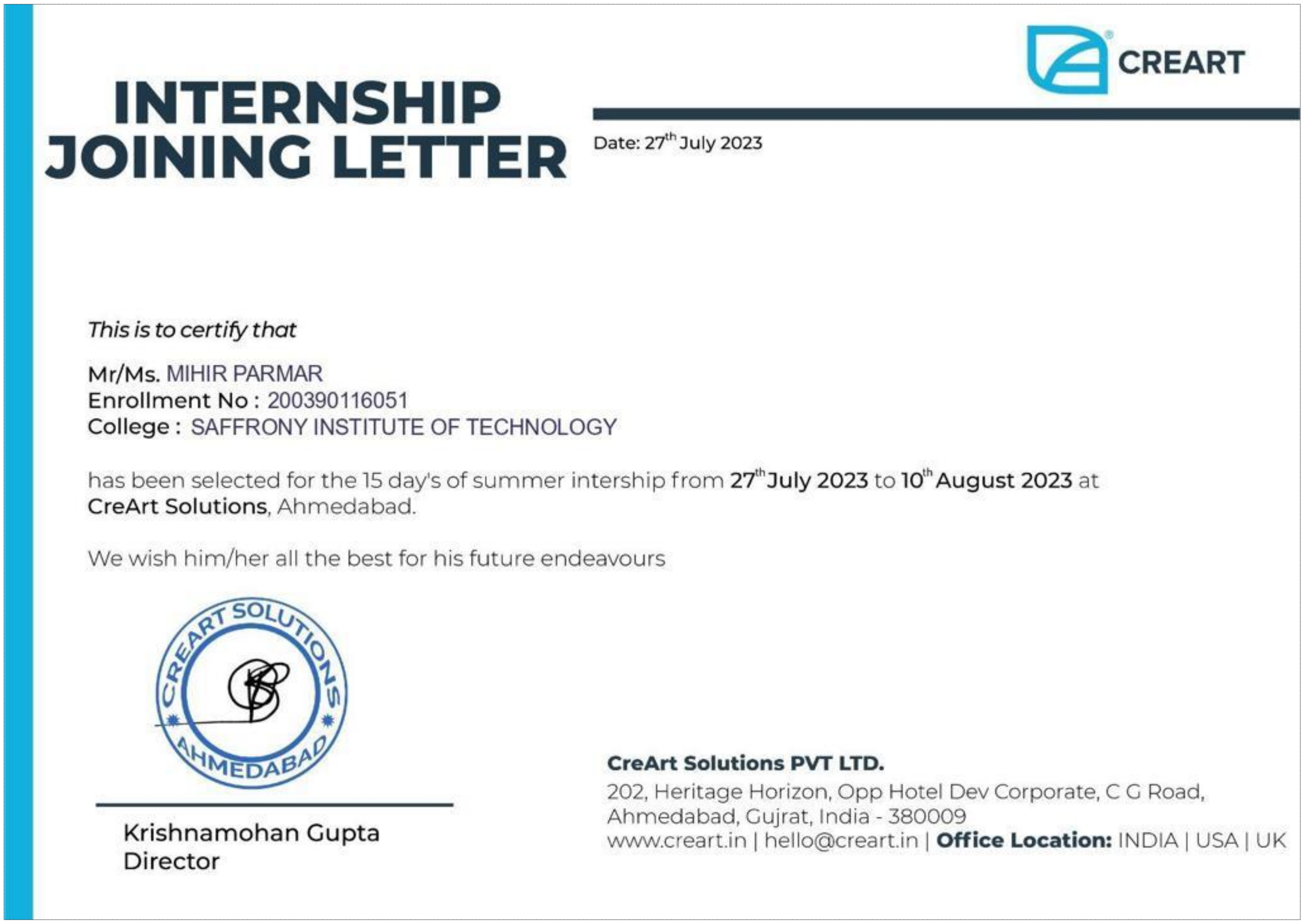
This is to certify that the project report submitted along with the project entitled Internship at **CREART SOLUTIONS PVT. LTD.** has been carried out by **Parmar Mihir Mehulbhai** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in Information Technology Engineering, 7th Semester of Gujarat Technological University, Ahmedabad during the academic year 2023-24.

Prof.	Prof. Akshay Kansara
Internal Guide	Head of Department

S.P.B. PATEL
ENGINEERING COLLEGE
SAFFRONY INSTITUTE OF TECHNOLOGY

GUJARAT TECHNOLOGICAL UNIVERSITY

JOINING LETTER



CREART

INTERNSHIP JOINING LETTER

Date: 27th July 2023

This is to certify that

Mr/Ms. MIHIR PARMAR
Enrollment No : 200390116051
College : SAFFRONY INSTITUTE OF TECHNOLOGY

has been selected for the 15 day's of summer intership from **27th July 2023** to **10th August 2023** at **CreArt Solutions**, Ahmedabad.

We wish him/her all the best for his future endeavours

CREART SOLUTIONS AHMEDABAD

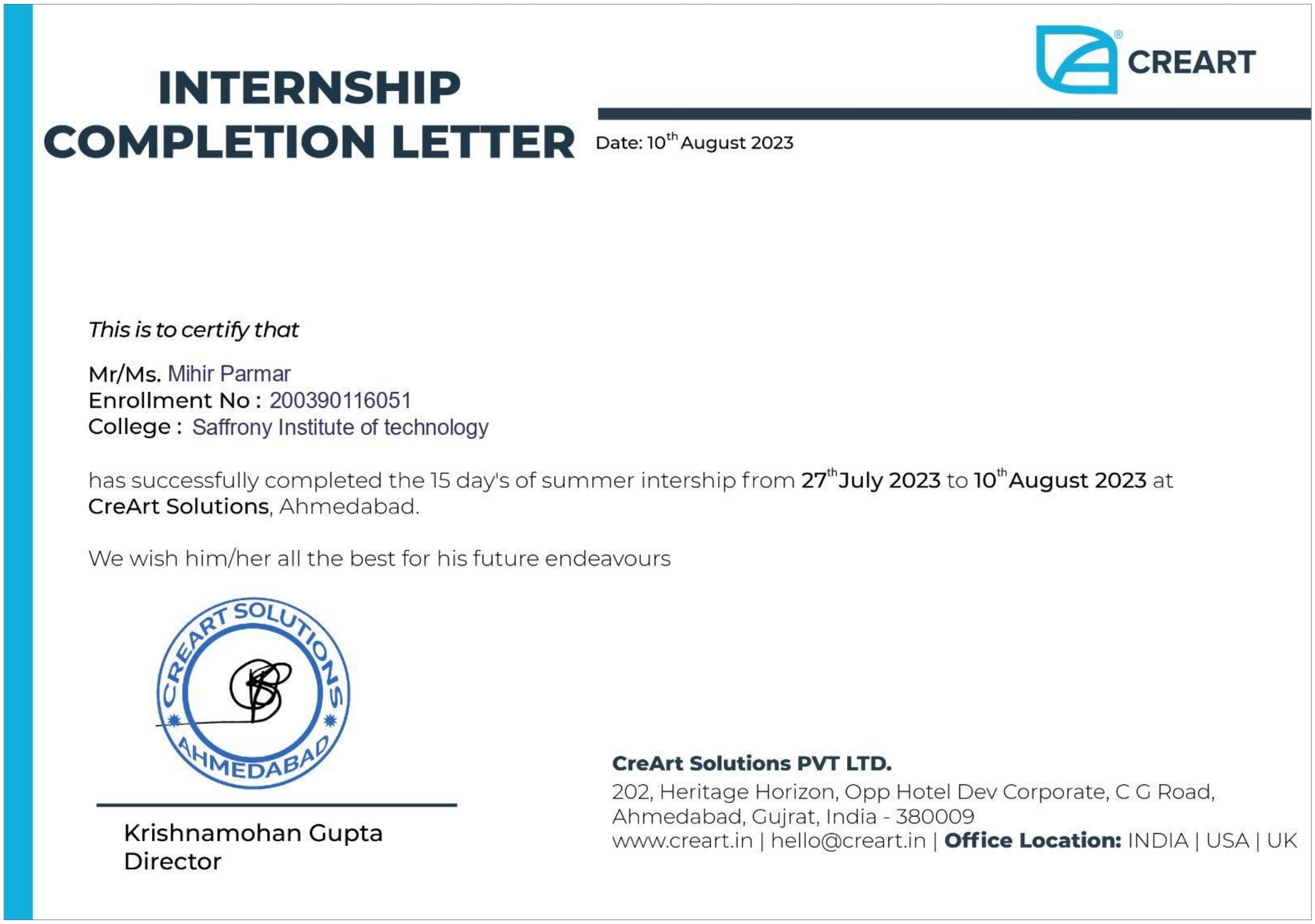
Krishnamohan Gupta
Director

CreArt Solutions PVT LTD.
202, Heritage Horizon, Opp Hotel Dev Corporate, C G Road,
Ahmedabad, Gujrat, India - 380009
www.creart.in | hello@creart.in | **Office Location:** INDIA | USA | UK

S.P.B. PATEL
ENGINEERING COLLEGE
SAFFRONY INSTITUTE OF TECHNOLOGY

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPLETION LETTER



CREART

INTERNSHIP COMPLETION LETTER

Date: 10th August 2023

This is to certify that

Mr/Ms. Mihir Parmar
Enrollment No : 200390116051
College : Saffrony Institute of technology

has successfully completed the 15 day's of summer intership from **27th July 2023** to **10th August 2023** at **CreArt Solutions**, Ahmedabad.

We wish him/her all the best for his future endeavours

CREART SOLUTIONS AHMEDABAD

Krishnamohan Gupta
Director

CreArt Solutions PVT LTD.
202, Heritage Horizon, Opp Hotel Dev Corporate, C G Road,
Ahmedabad, Gujrat, India - 380009
www.creart.in | hello@creart.in | **Office Location:** INDIA | USA | UK

S.P.B. PATEL
ENGINEERING COLLEGE
SAFFRONY INSTITUTE OF TECHNOLOGY

S.P.B. Patel Engineering College, Mehsana

Near Shanku's Water Park, Ahmedabad – Mehsana Highway, Linch, Gujarat

DECLARATION

I do declare that the Internship report submitted along with the Internship entitled on **NoticeHub** using **D'jango Python** at **CREART SOLUTIONS PVT. LTD.** submitted in partial fulfillment for the degree of Bachelor of Engineering in **Information Technology Engineering** to Gujarat Technological University, Ahmedabad.

I hereby warrant that the work I have presented does not breach any existing copyright acts.

Name of the Student	Sign of Student
Mihir Parmar	______________________

COMPANY PROFILE

CreArt Solutions is a privately owned venture of IT Solutions and IT Consultants formed in 2013. We always aspire to create a resistant future. We believe in bringing Business, People and Technology together in the way forward. CreArt is focused on rigorous development and comprehensive quality.

CreArt is dedicated towards perfection in every aspect. Professionalism is the main ingredient of CreArt. We strongly believe in delivering the best services to the clients till their satisfaction. Main objective of CreArt is to provide professional , qualitative , innovative and accessible services in every possible form.

Our objective is the strong collaboration between design, development, and delivering services on time, which benefits not only clients but also the communities in which the projects are implemented. More to help businesses succeed worldwide, we offer end-to-end development of web and mobile apps that integrate dynamic modern technologies like AR/VR Development, AI, Machine Learning and many more.

ACKNOWLEDGMENT

The internship experience I had with **CREART SOLUTIONS PVT. LTD.** was truly invaluable, providing me with a remarkable opportunity for learning and professional growth. I consider myself extremely fortunate to have been given this chance.

I would like to express my sincere gratitude to Mr.Alkesh kaba, the director and instructor at CREART SOLUTIONS PVT. LTD., for his guidance and mentorship throughout my internship. His insights and expertise were instrumental in shaping my learning experience. Additionally, I extend my heartfelt thanks to Krishnamohan Gupta, the CEO at CREART SOLUTIONS PVT. LTD., for her support and assistance.

This internship has marked a significant milestone in my career development, offering me a platform to acquire valuable skills and knowledge. I am committed to applying these newly acquired abilities in the most effective manner, while continuously striving to enhance and refine them. My dedication is geared towards achieving my ultimate career objectives.

ABSTRACT

NoticeHub is a web application by which we can send the Notice to all employees/students etc. it has basically two panels: Admin & User. We can create & update Notices from Admin Panel and directly show it to all Users on User Panel.

It is very easy process to perform task, we just required a network connection. First we have to connect XAMPP server then go to browser, open the admin Panel, Create new notice or Update existing notice & lastly click on the submit button so that User successfully receive the Notices without any failure. It was implementing using HTML & CSS for frontend and Django, Python & MySQL for backend.

List of Figures

TABLE OF CONTENT

Chapter:1 About the Company

1.1 Introduction of Company

Fig. 1.1 Company Logo

- CreArt is a privately owned venture of IT Solutions and IT Consultants formed in 2013. We always aspire to create a resistant future. We believe in bringing Business, People and Technology together in the way forward. CreArt is focused on rigorous development and comprehensive quality.

- CreArt is dedicated towards perfection in every aspect. Professionalism is the main ingredient of CreArt. They strongly believe in delivering the best services to the clients till their satisfaction.

- Main objective of CreArt is to provide professional , qualitative , innovative and accessible services in every possible form.We take care of development, deliveries, required resources and innovation.Our experienced development and consultancy will add value to mobile or web application development project.

- Our objective is the strong collaboration between design, development, and delivering services on time, which benefits not only clients but also the communities in which the projects are implemented. More to help businesses succeed worldwide, we offer end-to-end development of web and mobile apps that integrate dynamic modern technologies like AR/VR Development, AI, Machine Learning and many more.

- I would like to express my sincere gratitude to Mr.Alkesh kaba, the director and instructor at CREART SOLUTIONS PVT. LTD., for his guidance and mentorship throughout my internship. His insights and expertise were instrumental in shaping my learning experience.

Chapter:2 About the Domain & IDE

2.1 Introduction of Core Python

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development,

- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

- Core Python is a programming branch of Python. It is employed in the development of areas such as artificial intelligence and desktop apps. The core Python language includes basic commands and data structures such as tuple, list, and dictionary implementation to perform certain tasks

2.2 Introduction & Installation of Text Editor (Sublime 3)

- There is recommended system requirement for python and so many IDEs are available for python. They also told us the pros and cons of different IDEs.

- There are many software available for programming in Python like.. PyCharm, Notepad++, VScode, Sublime text 3, atom and NetBeans etc. I have installed PyCharm, VScode and Sublime text. mostly I am work on Sublime Text3.

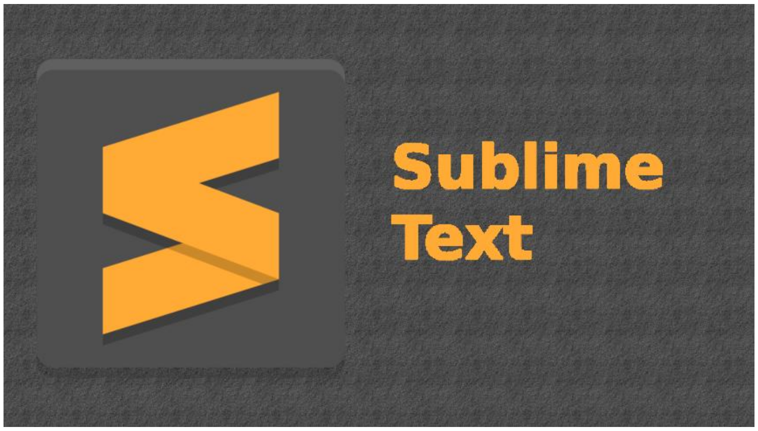
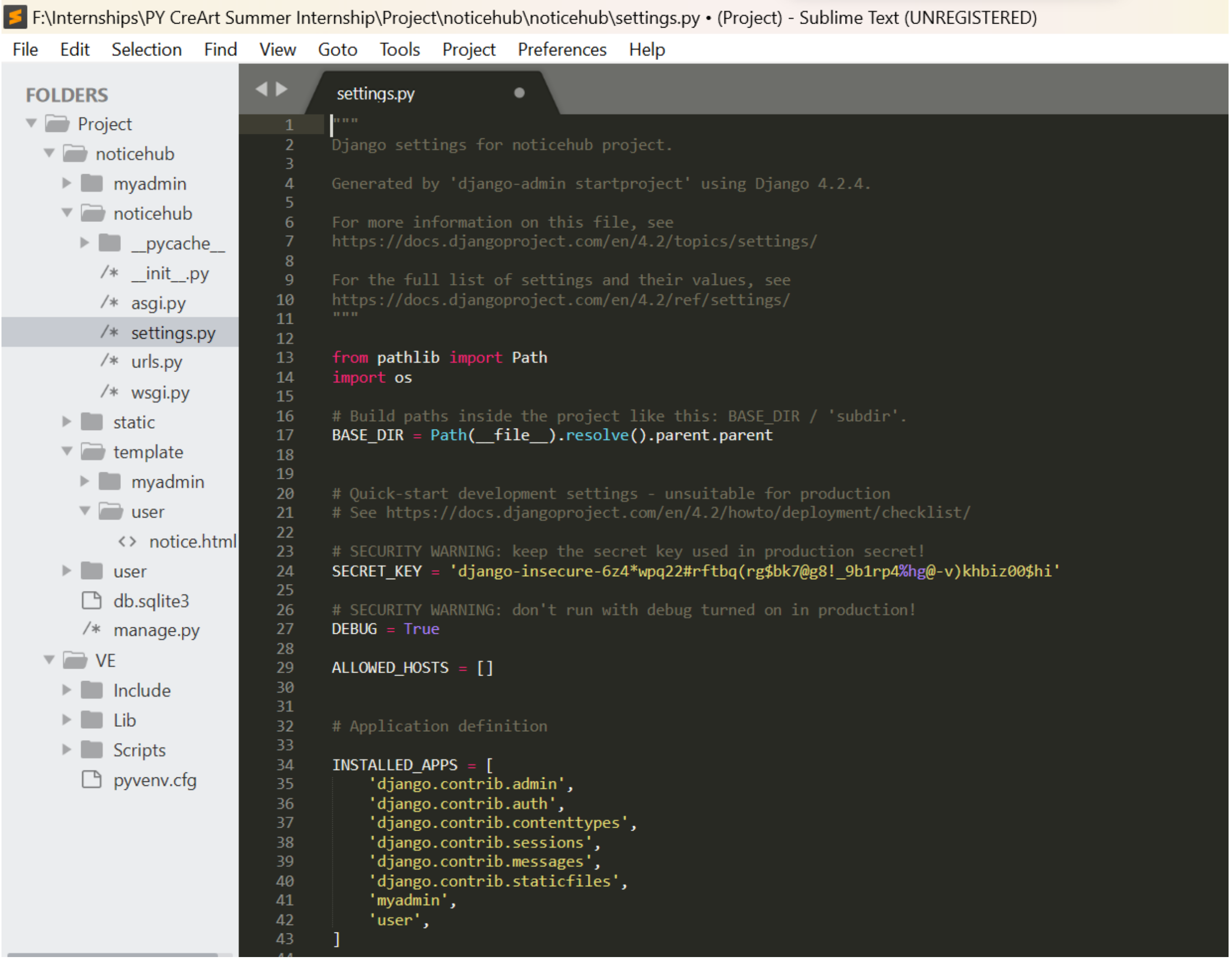


Fig. 2.2.1 Sublime Text 3 Logo

- We can directy Download Sublime Text 3 from https://www.sublimetext.com/3. And then directly Install it. After Installation Open Folder as shown below.



```
"""
Django settings for noticehub project.

Generated by 'django-admin startproject' using Django 4.2.4.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.2/ref/settings/
"""

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-6z4*wpq22#rftbq(rg$bk7@g8!_9b1rp4%hg@-v)khbiz00$hi'

# 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',
    'myadmin',
    'user',
]
```

Fig. 2.2.2 Folder in Sublime Text 3

- In Sublime, We can create a Folder and then create a python file with **.py** extension. Here we see the folder named **Project** and inside this folder I created another folder name **noticehub** and inside this I created many other folders and many other python files.

Chapter:3 Datatypes and Variables

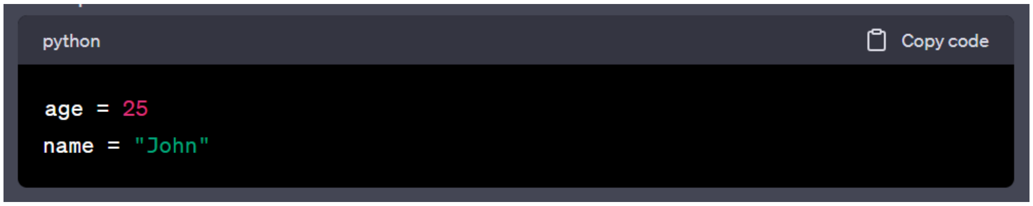
3.1 Variables & rules in Python

- In Python, variables are used to store and manipulate data values. They act as containers that hold various types of information, such as numbers, strings, lists, or more complex data structures.

- Certainly! In Python, variables serve as containers to store different types of data. They are like labels that help you refer to values easily. To create a variable, you simply give it a name and assign a value using the '=' symbol. Python is dynamically typed, so you don't need to specify the type of data the variable will hold.

- Variables can store integers (whole numbers), floats (decimal numbers), strings (text), Booleans (True or False), and more complex structures like lists (ordered collections), tuples (similar to lists but immutable), dictionaries (key-value pairs), and sets (unordered unique values).

- You can change a variable's value by assigning it a new value, and you can use variables in strings using techniques like f-strings. Just remember that variable names must follow certain rules and conventions, and they can have different scopes based on where they are defined, affecting their accessibility within your code.



```python
age = 25
name = "John"
```

Fig. 3.1.1 Variables in Python

3.2 Datatypes & Type casting in Python

- In Python, data types define the kind of values a variable can hold. Here's a quick overview of some common data types:

1. int: Represents integers (whole numbers), like 5 or -12.
2. float: Represents floating-point numbers (decimal numbers), such as 3.14 or -0.5.
3. str: Stands for string and represents text, like "Hello, world!".

4. bool: Represents Boolean values, True or False, often used for logical operations.
5. list: Represents an ordered collection of items, which can be of mixed types.
6. tuple: Similar to lists but immutable, meaning their contents can't be changed after creation.
7. dict : Stands for dictionary, representing a collection of key-value pairs for efficient data lookup.
8. set: Represents an unordered collection of unique items, useful for mathematical operations.
9. None Type: Represents the absence of a value, often used to indicate a variable with no assigned value.
10. complex: Represents complex numbers with a real and an imaginary part.

- Python's dynamic typing allows you to change the type of a variable on the fly, making it flexible but also requiring caution. Understanding these data types is crucial for effective programming and data manipulation.

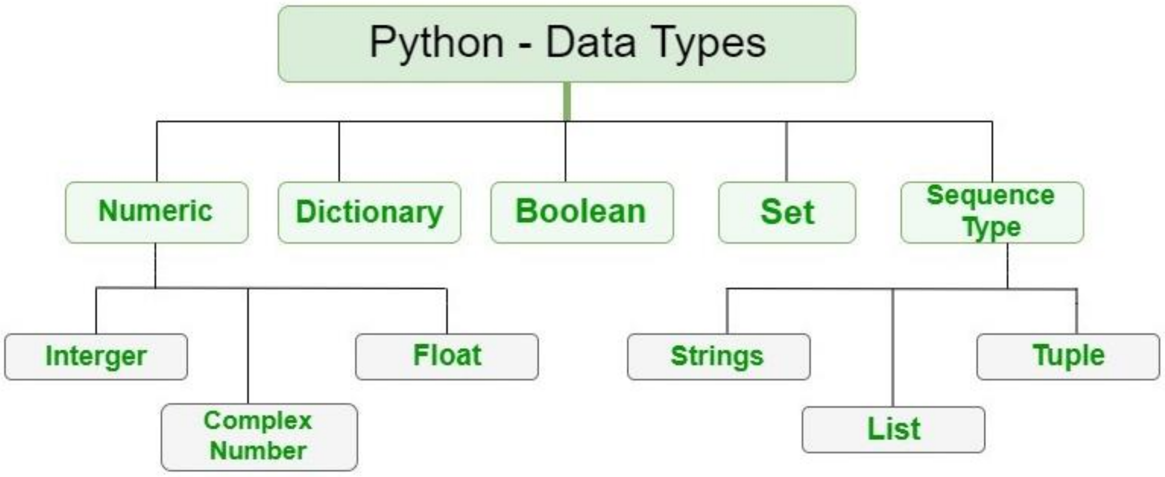
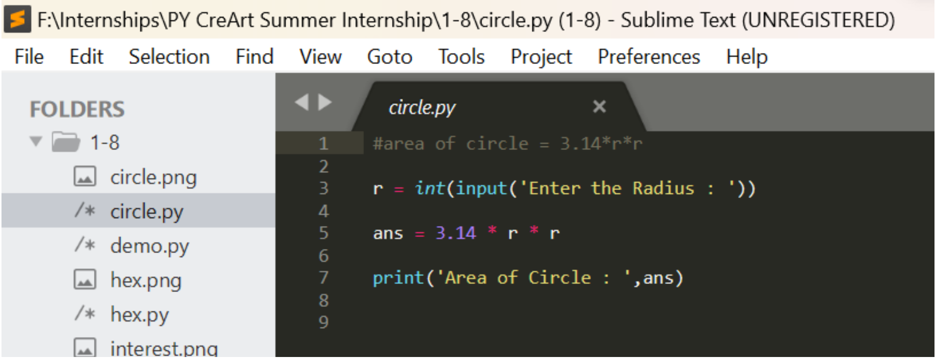


Fig. 3.2.1 Datatypes in Python

Chapter:4 Problem solving Examples

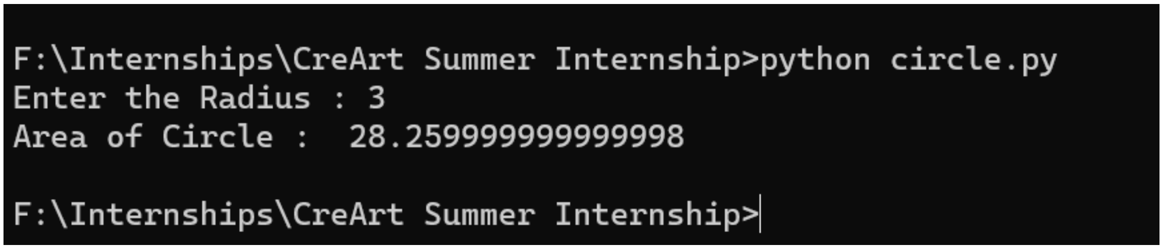
- Here are some Mathematical Examples solved by Python Programming..

4.1 Area Of Circle :



```python
#area of circle = 3.14*r*r

r = int(input('Enter the Radius : '))

ans = 3.14 * r * r

print('Area of Circle : ',ans)
```

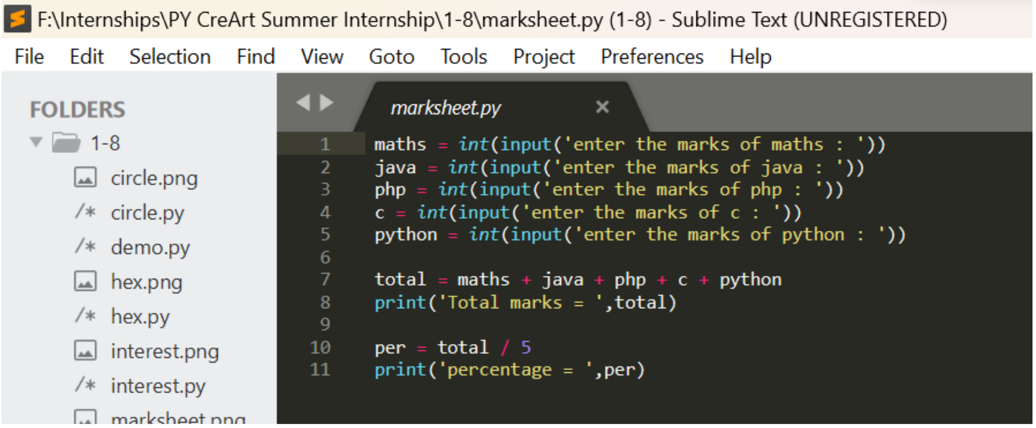
Fig. 4.1.1 Example 1



```
F:\Internships\CreArt Summer Internship>python circle.py
Enter the Radius : 3
Area of Circle :  28.259999999999998

F:\Internships\CreArt Summer Internship>
```

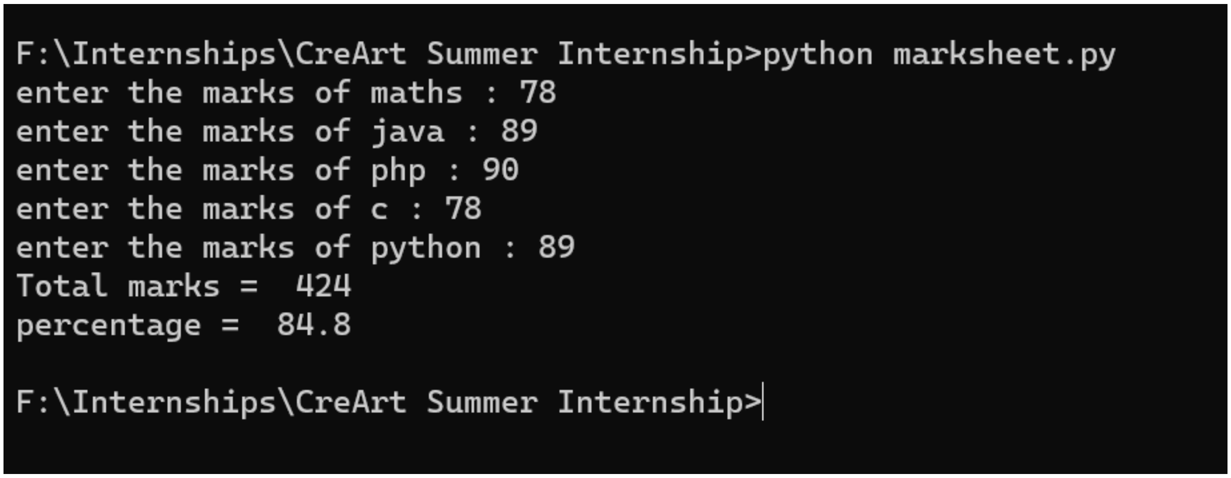
Fig. 4.1.2 Output 1

4.2 Marksheet :



```python
maths = int(input('enter the marks of maths : '))
java = int(input('enter the marks of java : '))
php = int(input('enter the marks of php : '))
c = int(input('enter the marks of c : '))
python = int(input('enter the marks of python : '))

total = maths + java + php + c + python
print('Total marks = ',total)

per = total / 5
print('percentage = ',per)
```

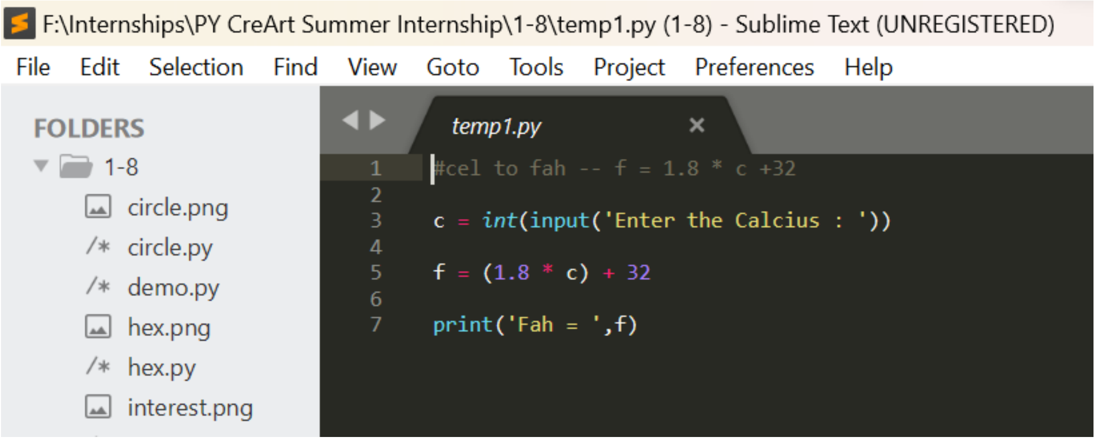
Fig. 4.2.1 Example 2



```
F:\Internships\CreArt Summer Internship>python marksheet.py
enter the marks of maths : 78
enter the marks of java : 89
enter the marks of php : 90
enter the marks of c : 78
enter the marks of python : 89
Total marks =  424
percentage =  84.8

F:\Internships\CreArt Summer Internship>
```

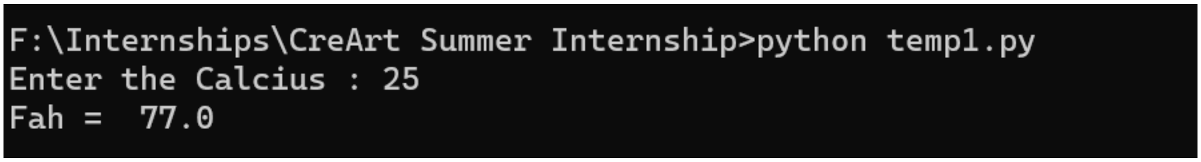
Fig. 4.2.2 Output 2

4.3 Temperature Cel to Fah :



```
#cel to fah -- f = 1.8 * c +32

c = int(input('Enter the Calcius : '))

f = (1.8 * c) + 32

print('Fah = ',f)
```

Fig. 4.3.1 Example 3



```
F:\Internships\CreArt Summer Internship>python temp1.py
Enter the Calcius : 25
Fah =  77.0
```

Fig. 4.3.2 Output 3

- There are many more examples similar like these saved in system with python file and it’s Output screenshot.

Chapter:5 Functions in Python

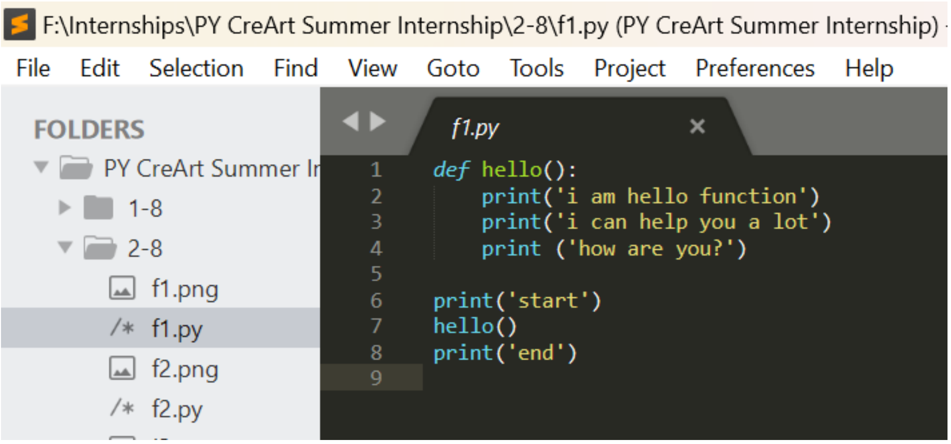
5.1 Introduction and Uses of Function :

- A Function in Python is a reusable piece of code that performs specific tasks. It's defined using the **def** keyword, followed by a name and optional parameters in parentheses. Functions are called by their name with parentheses, and they can return values using the return statement.

- They help organize code, improve reusability, and facilitate code comprehension. Function parameters have local scope, while variables outside functions have global scope. Functions can take both positional and keyword arguments, and default values can be set for parameters.

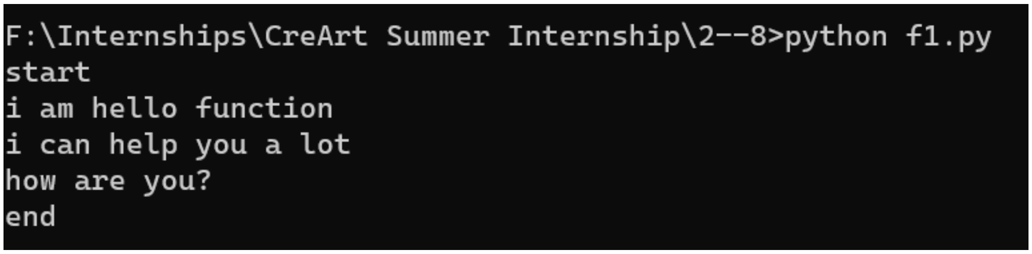
- In python def is used for creating function like ...

  { def FunctionName (arg1, arg2, ...):
  statement1
  statement2 }



```python
def hello():
    print('i am hello function')
    print('i can help you a lot')
    print ('how are you?')

print('start')
hello()
print('end')
```

Fig. 5.1.1 Example of Function



```
F:\Internships\CreArt Summer Internship\2--8>python f1.py
start
i am hello function
i can help you a lot
how are you?
end
```

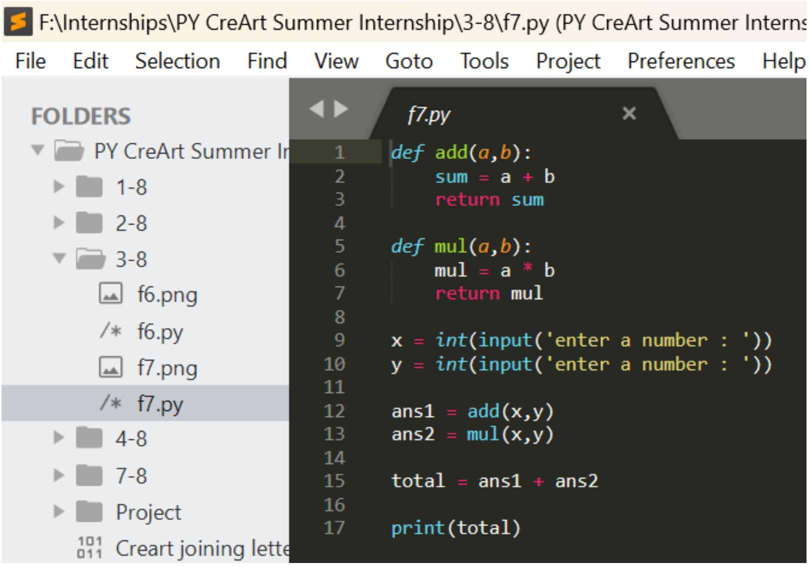
Fig. 5.1.2 Output of Function

Chapter:6 RETURN keyword in Python

6.1 Get Input from Users using RETURN Keyword:

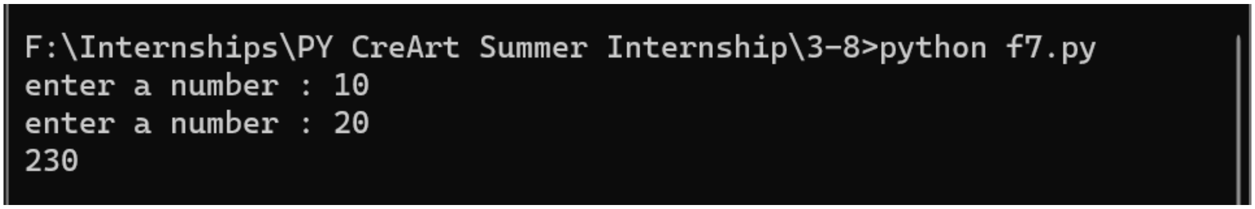
- In Python, the **return** keyword is used within a function to send a value back to the caller. When a function is executed and encounters a **return** statement, it immediately ends the function's execution and provides the specified value as the function's output.

- This returned value can then be assigned to a variable or used in further calculations. The return statement is essential for functions that need to provide results or computed values to the rest of the program. If a **return** statement is omitted or used without a value, the function returns **None** by default.


F:\Internships\PY CreArt Summer Internship\3-8\f7.py (PY CreArt Summer Interns
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
PY CreArt Summer In
1-8
2-8
3-8
f6.png
f6.py
f7.png
f7.py
4-8
7-8
Project
Creart joining lette
f7.py

```python
def add(a,b):
    sum = a + b
    return sum

def mul(a,b):
    mul = a * b
    return mul

x = int(input('enter a number : '))
y = int(input('enter a number : '))

ans1 = add(x,y)
ans2 = mul(x,y)

total = ans1 + ans2

print(total)
```

Fig. 6.1.1 Example of RETURN keyword



```
F:\Internships\PY CreArt Summer Internship\3-8>python f7.py
enter a number : 10
enter a number : 20
230
```

Fig. 6.1.2 Output of RETURN keyword

Chapter:7 D'jango Framework

7.1 Introduction of D'jango Framework :

- D'jango is a high-level python web framework that enables rapid development of secure and maintainable websites. D'jango is used because of its versatility, Scalability, Security (Using SQL Injection, CSRF Protection), Pre-Requisite (ORM (Object Relationships Mapper)), Large Library Support, Free and Open Source.

- It is mostly used in web-based application and servers side websites and particular the mechanics of client-server interactions in websites. Web app means User can perform action on browser and also can use in mobile and PC both.

- Why D'jango :-
  - ✓ Scalable
  - ✓ Security
  - ✓ Performance
  - ✓ Library
  - ✓ Open source and free
  - ✓ Versatile:- multiple tasks

7.2 Installation of D'jango Framework and Project Setup Steps:

- There are steps following for create Django project: -

  1. Create Virtual Environment and Activate.
  2. Download Django and install it.
  3. Create project.
  4. Run on Server.
  5. Install MySQL client library.
  6. Then migrate the database using **python mange.py migrate**.
  7. Create superuser (Admin) by **python manage.py createsuperuser.**
  8. In project run server using **python manage.py runserver** command here one local host address was displayed: - http:// 127.0.0.1.8000.

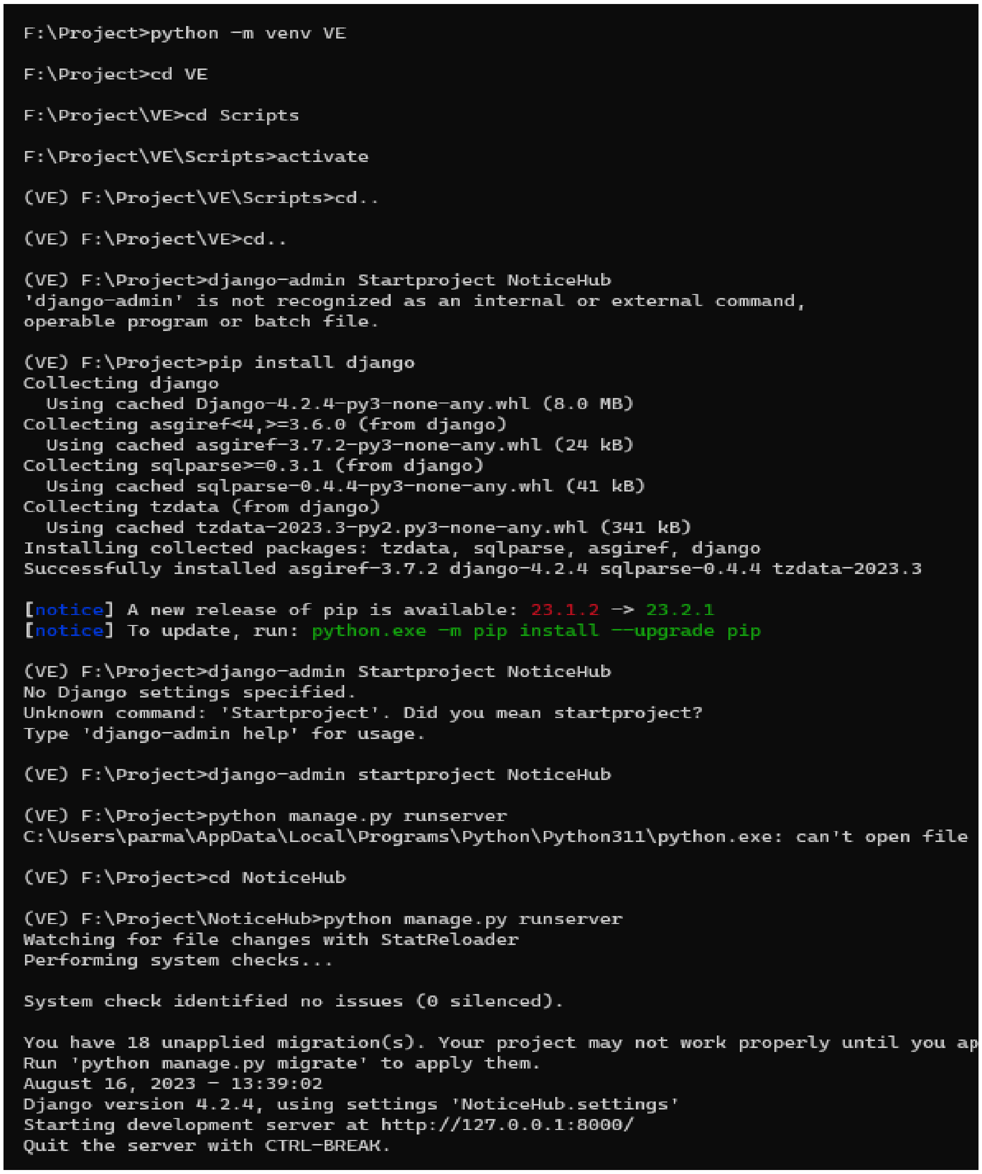
- **Project Setup :**

1) First create & Activate Virtual Environment using Python by following command:
Python –m venv VE & VE>Scripts>activate

2) Install Django using following command:
pip install Django

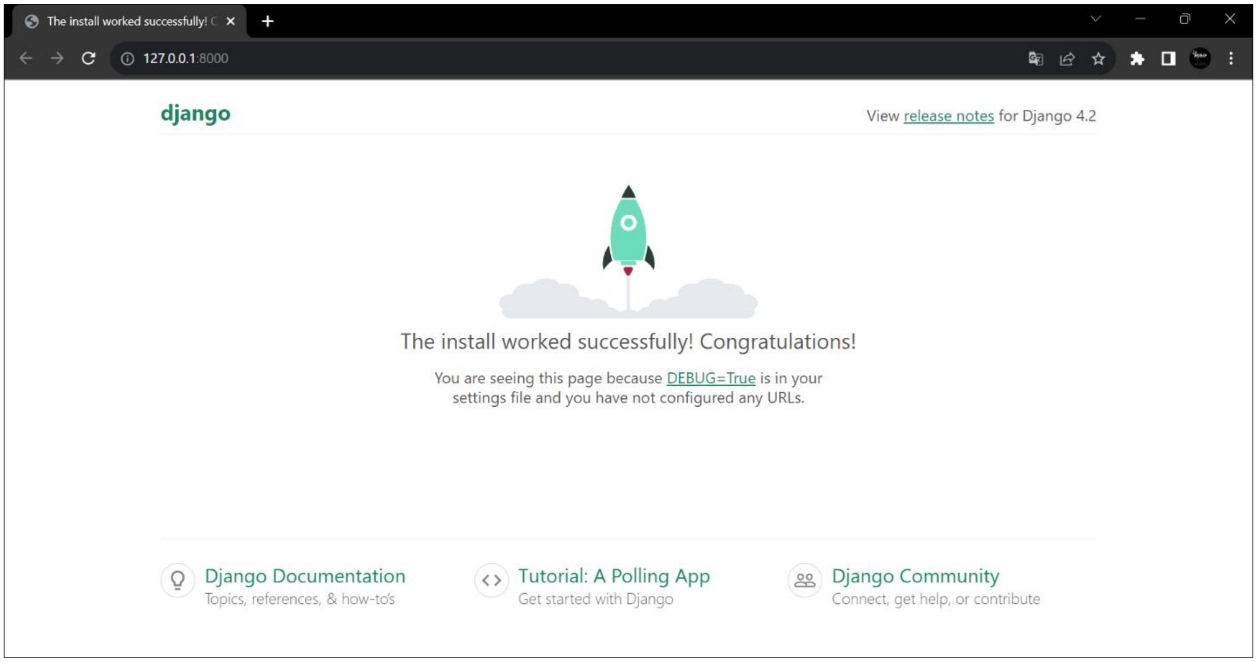
3) Create project.
Project>Django-admin startproject NoticeHub

4) Run on Server.
Project>python manage.py runserver



```
F:\Project>python -m venv VE

F:\Project>cd VE

F:\Project\VE>cd Scripts

F:\Project\VE\Scripts>activate

(VE) F:\Project\VE\Scripts>cd..

(VE) F:\Project\VE>cd..

(VE) F:\Project>django-admin Startproject NoticeHub
'django-admin' is not recognized as an internal or external command,
operable program or batch file.

(VE) F:\Project>pip install django
Collecting django
  Using cached Django-4.2.4-py3-none-any.whl (8.0 MB)
Collecting asgiref<4,>=3.6.0 (from django)
  Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting tzdata (from django)
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.4 sqlparse-0.4.4 tzdata-2023.3

[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(VE) F:\Project>django-admin Startproject NoticeHub
No Django settings specified.
Unknown command: 'Startproject'. Did you mean startproject?
Type 'django-admin help' for usage.

(VE) F:\Project>django-admin startproject NoticeHub

(VE) F:\Project>python manage.py runserver
C:\Users\parma\AppData\Local\Programs\Python\Python311\python.exe: can't open file

(VE) F:\Project>cd NoticeHub

(VE) F:\Project\NoticeHub>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you ap
Run 'python manage.py migrate' to apply them.
August 16, 2023 - 13:39:02
Django version 4.2.4, using settings 'NoticeHub.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig. 7.2.1 Setup


The install worked successfully! C
127.0.0.1:8000
django
View release notes for Django 4.2

The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Django Documentation
Topics, references, & how-to's
Tutorial: A Polling App
Get started with Django
Django Community
Connect, get help, or contribute

Fig. 7.2.2 Setup Output

Chapter:8 & 9 Complete Project Setup

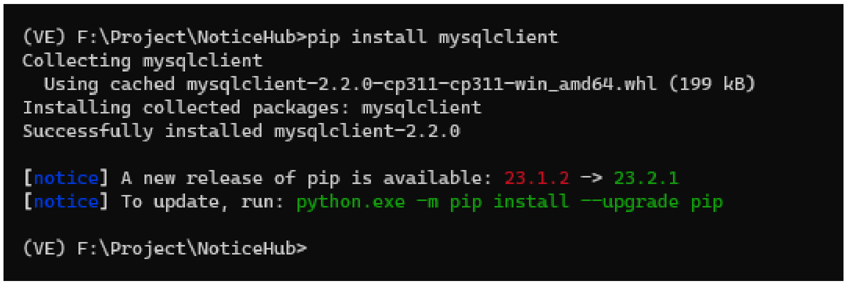
8-9.1 Complete Project Setup Steps:

5) Install MySQL client library:
pip install mysqlclient

6) Change in Setting.py file : SQLite to SQL database
Then migrate the database using:
python mange.py migrate

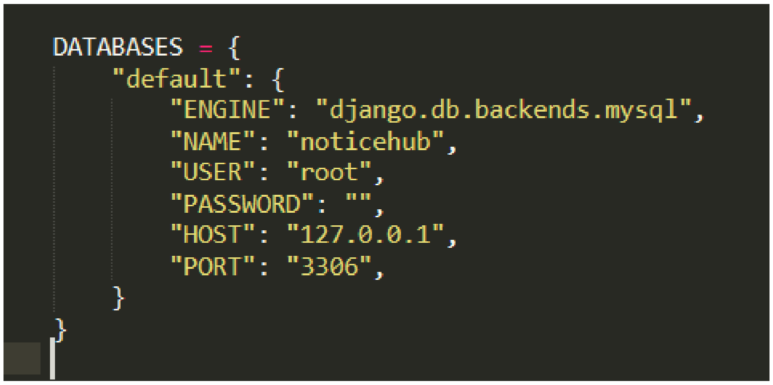
7) Create superuser (Admin) using:
python manage.py createsuperuser

8) Now run server using **python manage.py runserver** command here one local host address was displayed: - http:// 127.0.0.1.8000.



```
(VE) F:\Project\NoticeHub>pip install mysqlclient
Collecting mysqlclient
  Using cached mysqlclient-2.2.0-cp311-cp311-win_amd64.whl (199 kB)
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.2.0

[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(VE) F:\Project\NoticeHub>
```

Fig. 8-9.1.1 Install mysqlclient



```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.mysql",
        "NAME": "noticehub",
        "USER": "root",
        "PASSWORD": "",
        "HOST": "127.0.0.1",
        "PORT": "3306",
    }
}
```

Fig. 8-9.1.2 Change in database

- Once, the Changes are done in Setting.py (SQLite to SQL Database), Now we have to Create 10 tables by default with the help of Migrate command as shown in step number 6. Output of it as shown in Fig. 8-9.1.3

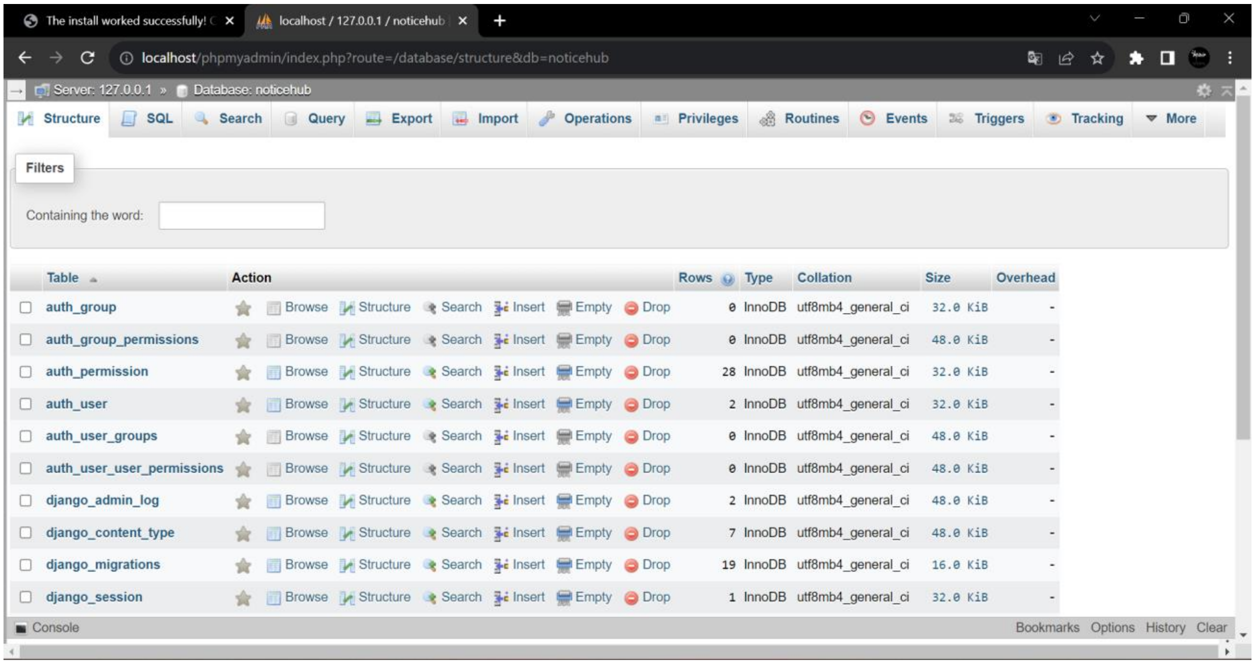

The install worked successfully!
localhost / 127.0.0.1 / noticehub
localhost/phpmyadmin/index.php?route=/database/structure&db=noticehub
Server: 127.0.0.1 » Database: noticehub
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
More
Filters
Containing the word:
Console
Bookmarks
Options
History
Clear

Table	Action							Rows	Type	Collation	Size	Overhead
□ auth_group	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□ auth_group_permissions	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□ auth_permission	★	Browse	Structure	Search	Insert	Empty	Drop	28	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□ auth_user	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□ auth_user_groups	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□ auth_user_user_permissions	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□ django_admin_log	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□ django_content_type	★	Browse	Structure	Search	Insert	Empty	Drop	7	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□ django_migrations	★	Browse	Structure	Search	Insert	Empty	Drop	19	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□ django_session	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-

Fig. 8-9.1.3 Migrate command output

- Now, we have to create Two superuser named Admin and User. once, the superusers are created in Noticehub, we have to check it can be run on server or not.

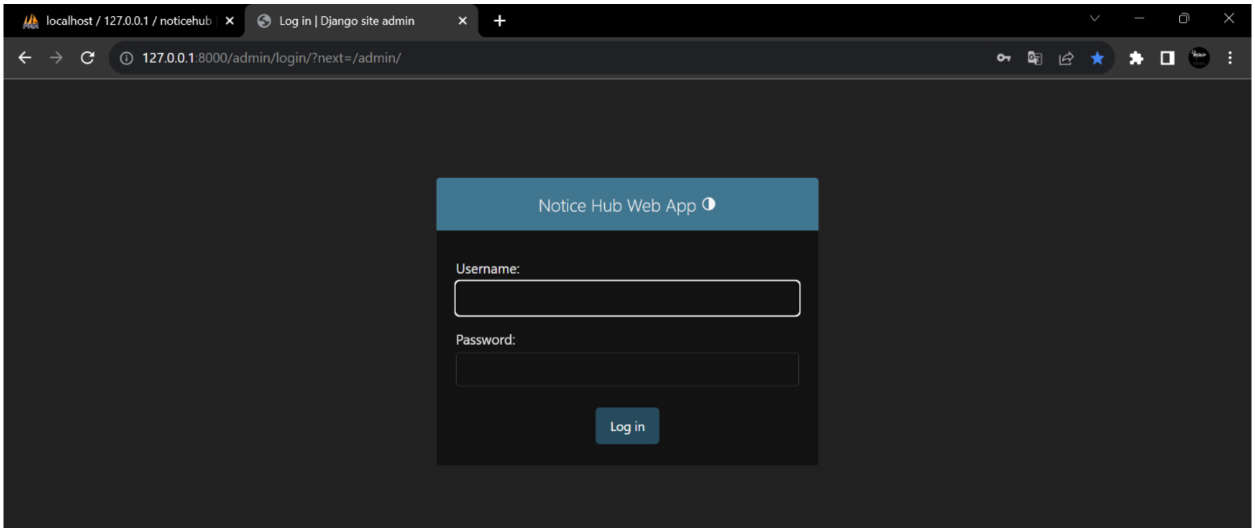
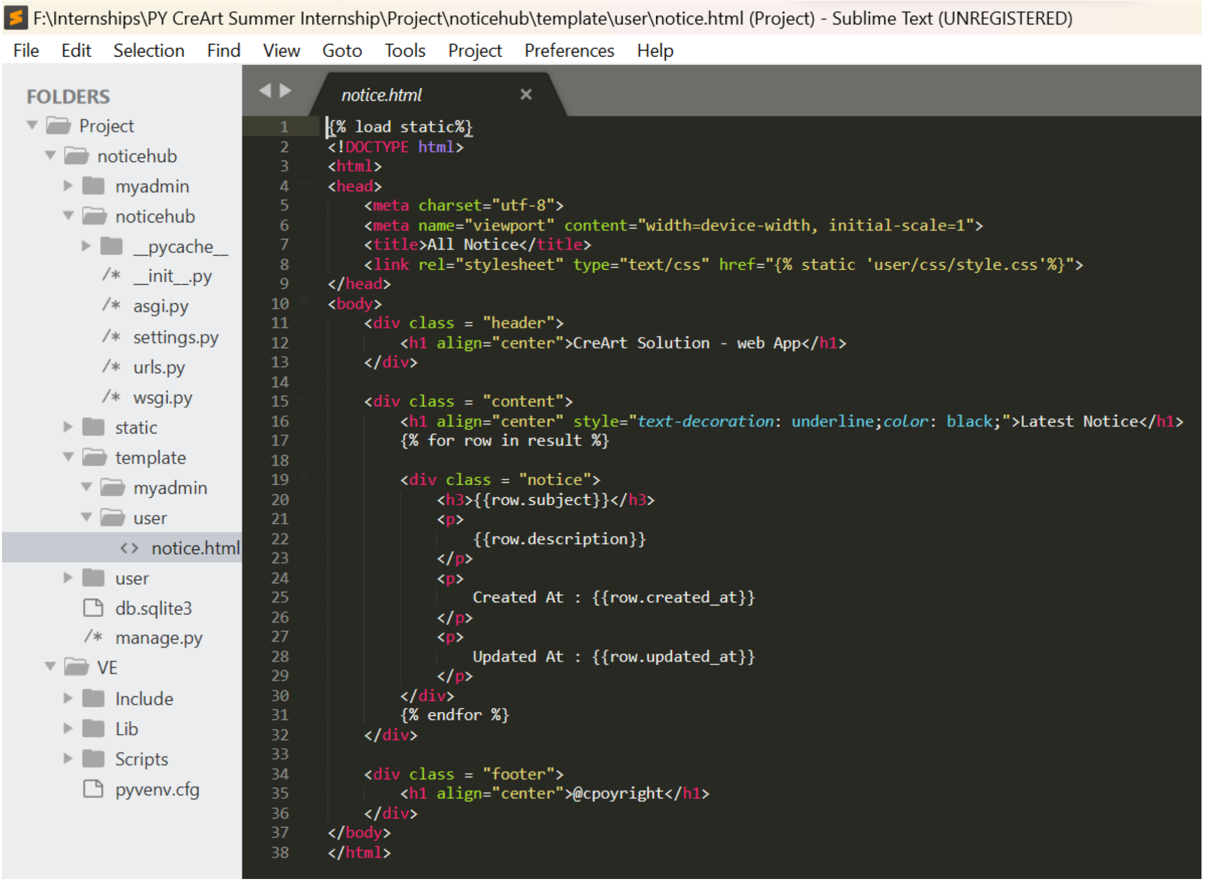

localhost / 127.0.0.1 / noticehub
Log in | Django site admin
127.0.0.1:8000/admin/login/?next=/admin/
Notice Hub Web App
Username:
Password:
Log in

Fig. 8-9.1.4 Admin panel

Chapter:10 Frontend of NoticeHub

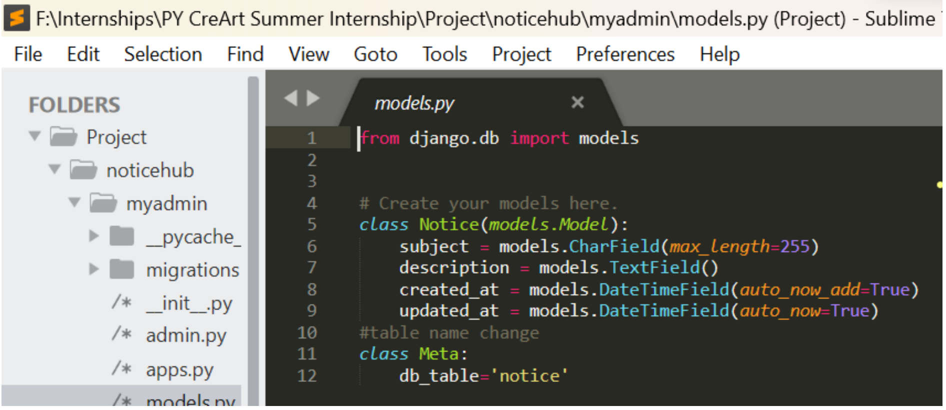
10.1 Frontend using D’jango Framework (using HTML) :

- Till 9th chapter, Our Project Setup is done. So now we go to Sublime Text 3 and make HTML file in User side folder. So that we can print notices on the web browser from our Project NoticeHub. For that we have to make **notice.html** file in **template** folder.



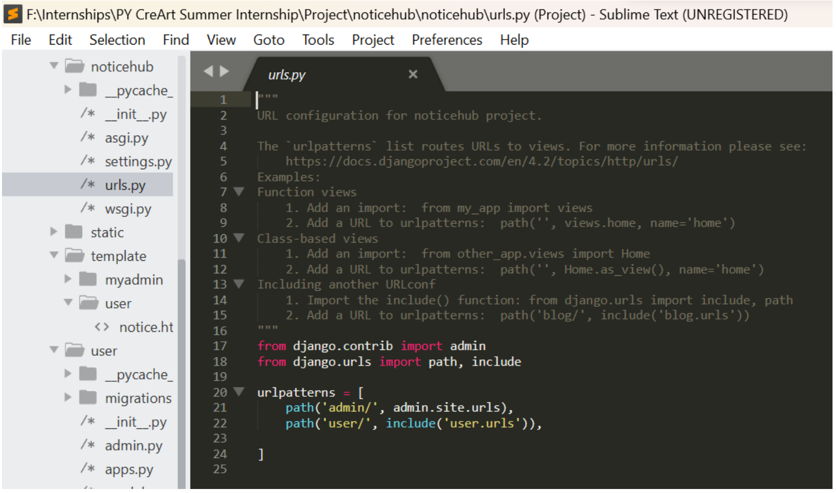
```html
{% load static%}
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>All Notice</title>
    <link rel="stylesheet" type="text/css" href="{% static 'user/css/style.css'%}">
</head>
<body>
    <div class = "header">
        <h1 align="center">CreArt Solution - web App</h1>
    </div>

    <div class = "content">
        <h1 align="center" style="text-decoration: underline;color: black;">Latest Notice</h1>
        {% for row in result %}

        <div class = "notice">
            <h3>{{row.subject}}</h3>
            <p>
                {{row.description}}
            </p>
            <p>
                Created At : {{row.created_at}}
            </p>
            <p>
                Updated At : {{row.updated_at}}
            </p>
        </div>
        {% endfor %}
    </div>

    <div class = "footer">
        <h1 align="center">@cpoyright</h1>
    </div>
</body>
</html>
```

Fig. 10.1.1 notice.html


F:\Internships\PY CreArt Summer Internship\Project\noticehub\myadmin\models.py (Project) - Sublime
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Project
noticehub
myadmin
__pycache_
migrations
__init__.py
admin.py
apps.py
models.py
models.py

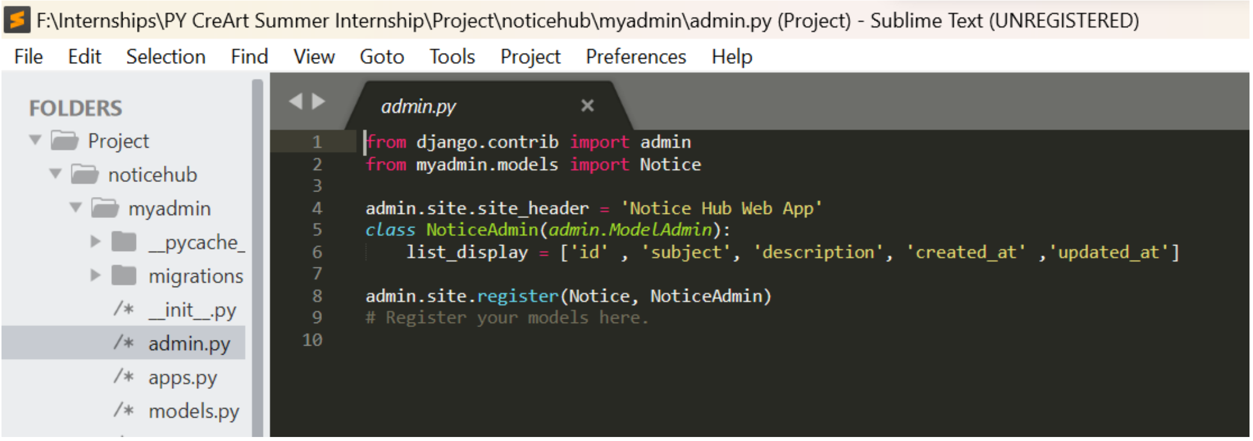
```python
from django.db import models


# Create your models here.
class Notice(models.Model):
    subject = models.CharField(max_length=255)
    description = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
#table name change
class Meta:
    db_table='notice'
```

Fig. 10.1.2 models.py


F:\Internships\PY CreArt Summer Internship\Project\noticehub\noticehub\urls.py (Project) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
noticehub
__pycache_
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
static
template
myadmin
user
notice.ht
user
__pycache_
migrations
__init__.py
admin.py
apps.py
urls.py

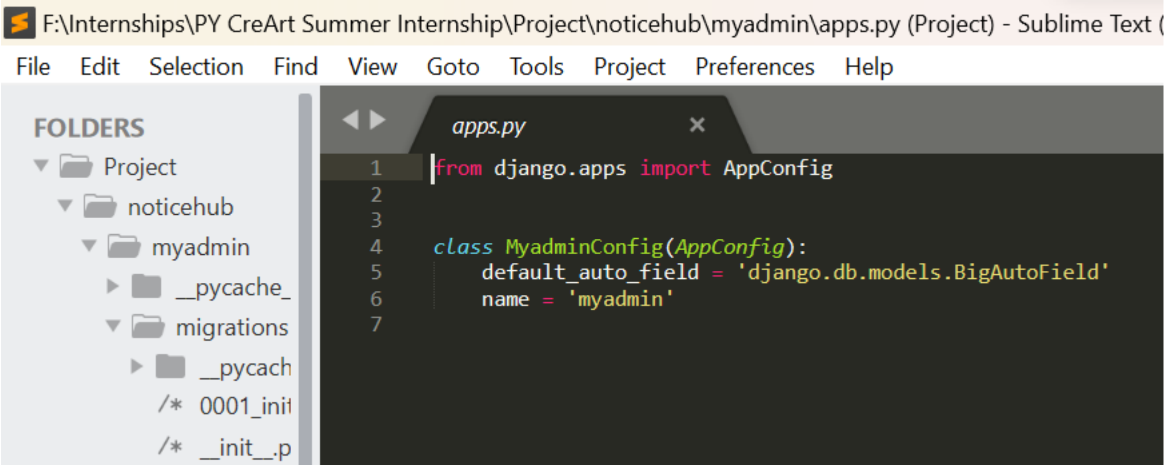
```python
"""
URL configuration for noticehub project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.2/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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('user/', include('user.urls')),

]

```

Fig. 10.1.3 urls.py


F:\Internships\PY CreArt Summer Internship\Project\noticehub\myadmin\admin.py (Project) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Project
noticehub
myadmin
__pycache_
migrations
__init__.py
admin.py
apps.py
models.py
admin.py

```python
from django.contrib import admin
from myadmin.models import Notice

admin.site.site_header = 'Notice Hub Web App'
class NoticeAdmin(admin.ModelAdmin):
    list_display = ['id' , 'subject', 'description', 'created_at' ,'updated_at']

admin.site.register(Notice, NoticeAdmin)
# Register your models here.

```

Fig. 10.1.4 admin.py


F:\Internships\PY CreArt Summer Internship\Project\noticehub\myadmin\apps.py (Project) - Sublime Text (
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Project
noticehub
myadmin
__pycache_
migrations
__pycach
0001_init
__init__.p
apps.py

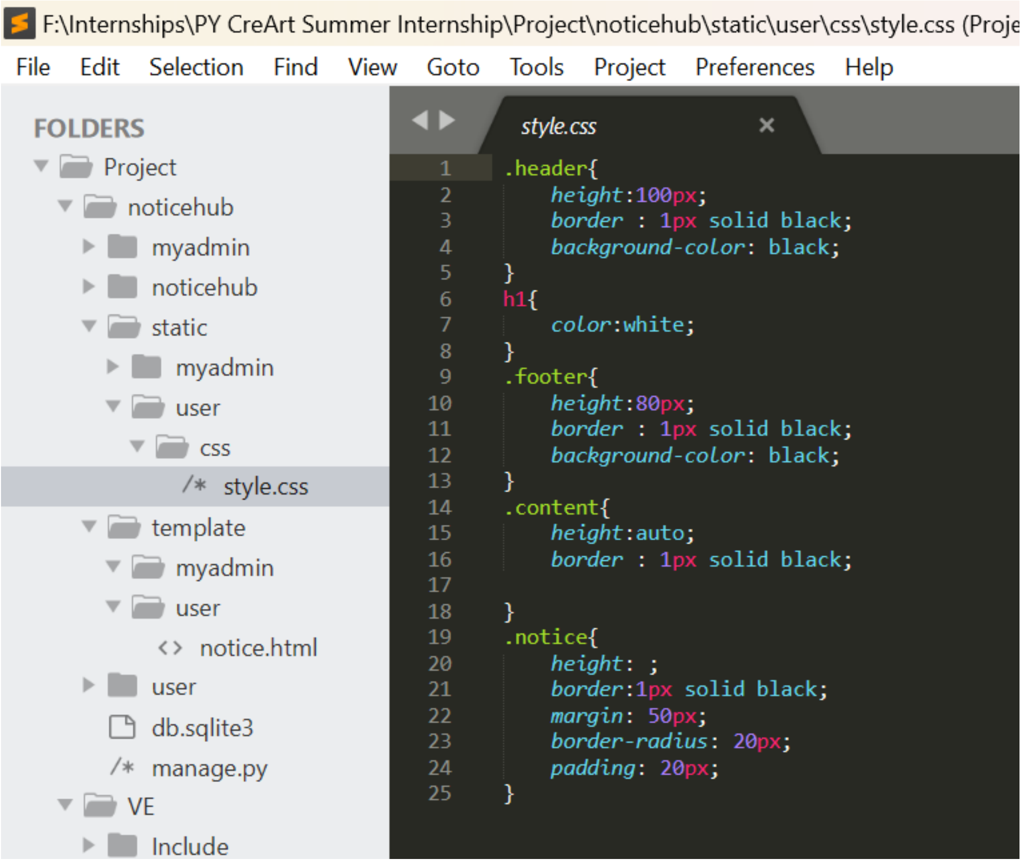
```python
from django.apps import AppConfig


class MyadminConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'myadmin'

```

Fig. 10.1.5 apps.py

Chapter:11 Apply CSS on Frontend

11.1 Applying CSS on Frontend and Complete the Frontend :

- Till 10th chapter, Our Project’s HTML Format Frontend is done. So now we have to decorate or style the frontend with the help of CSS. So that we can make a better GUI for our Project NoticeHub. For that we have to make **style.css** file in **static** folder.



```css
.header{
    height:100px;
    border : 1px solid black;
    background-color: black;
}
h1{
    color:white;
}
.footer{
    height:80px;
    border : 1px solid black;
    background-color: black;
}
.content{
    height:auto;
    border : 1px solid black;

}
.notice{
    height: ;
    border:1px solid black;
    margin: 50px;
    border-radius: 20px;
    padding: 20px;
}
```

Fig. 11.1.1 style.css

- So till now, we completed whole Frontend of our Project NoticeHub. Now, it’s time to run the project on the server and checks for the result.

Chapter:12 NoticeHub Project

12.1 Overview of NoticeHub :

- NoticeHub is a web application by which we can send the Notice to all employees/students etc. it has basically two panels: Admin & User. We can create & update Notices from Admin Panel and directly show it to all Users on User Panel.

- It is very easy process to perform task, we just required a network connection. First we have to connect XAMPP server then go to browser, open the admin Panel, Create new notice or Update existing notice & lastly click on the submit button so that User successfully receive the Notices without any failure. It was implementing using HTML & CSS for frontend and Django, Python & MySQL for backend.

- **Technology Used :**

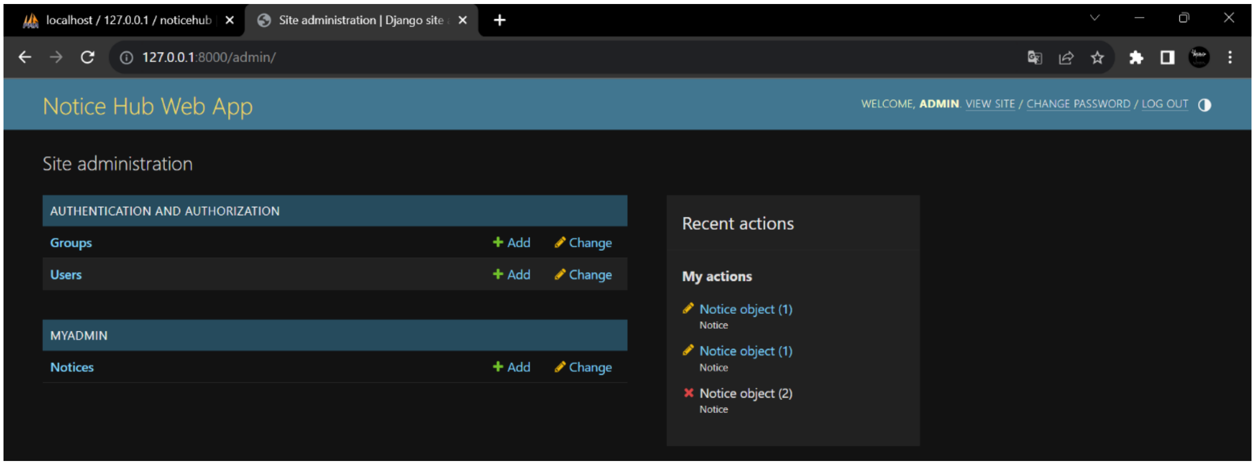
  - ➢ Frontend :
    - ⇨ HTML : HTML is used to create and save web document.
    - ⇨ CSS : Create attractive Layout and design the web page.
  - ➢ Backend :
    - ⇨ Django : Django is a Python based web framework that follows the model-template-views architectural pattern.
    - ⇨ MySQL : MySQL is a database, widely used for accessing querying and managing databases.

12.2 Run the NoticeHub :

- Here we make total two panels : Admin & User.

  - ➢ Admin panel : It's a basic Admin panel, Where Superusers can log in and make or update the notices. Also there are features for manage the selective notices. For go to Admin panel : **http://127.0.0.1:8000/admin/**

  - ➢ User panel : It's a basic User panel, Where all kinds of users like Employees or Students can see new notices and updated notices also. Here they can get User Friendly UI of NoticeHub. For go to User panel : **http://127.0.0.1:8000/user/notice**


localhost / 127.0.0.1 / noticehub
Site administration | Django site
127.0.0.1:8000/admin/
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Notice Hub Web App

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

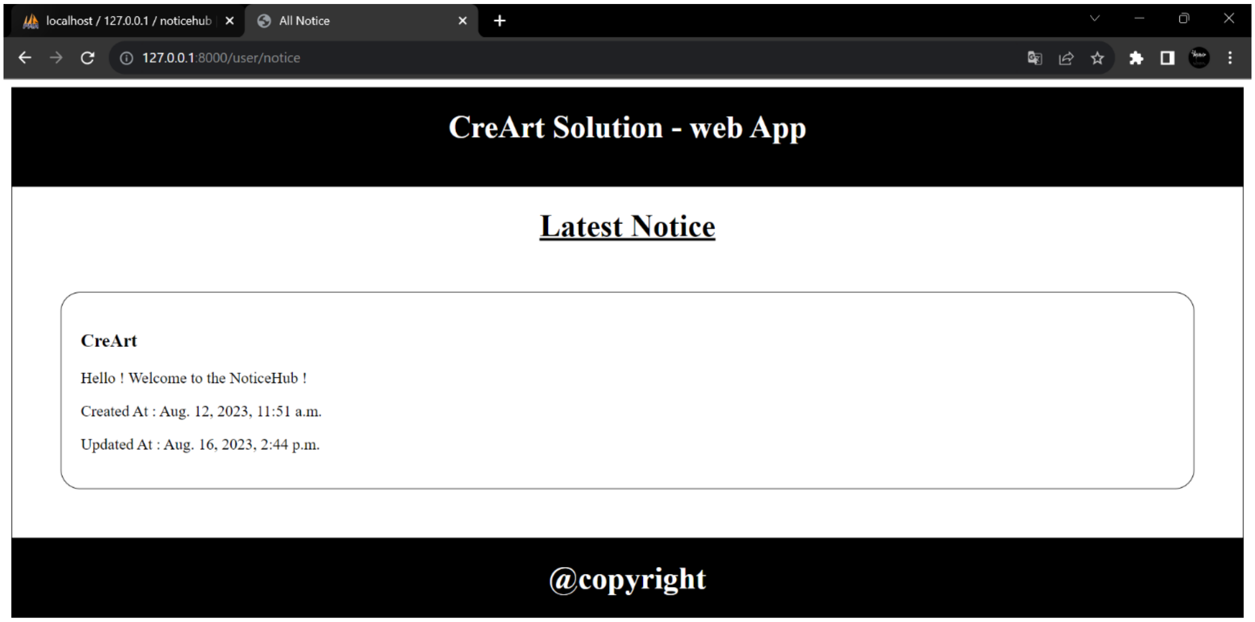
MYADMIN		
Notices	+ Add	Change

Recent actions

My actions

- Notice object (1)
Notice
- Notice object (1)
Notice
- ✗ Notice object (2)
Notice

Fig. 12.2.1 Admin Panel



CreArt Solution - web App

Latest Notice

CreArt

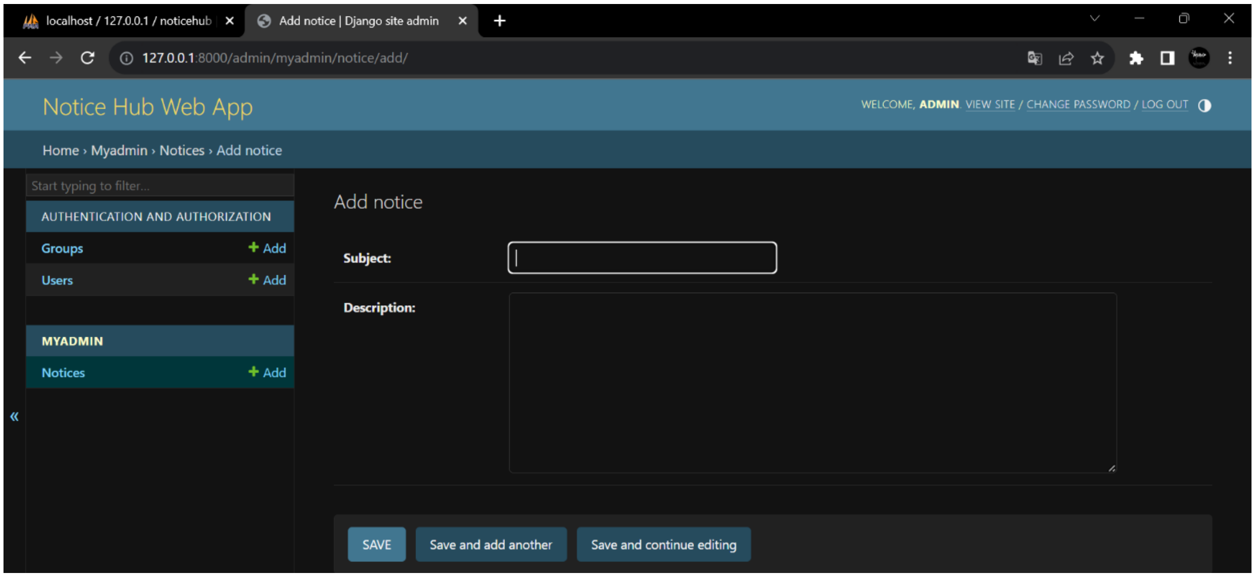
Hello ! Welcome to the NoticeHub !

Created At : Aug. 12, 2023, 11:51 a.m.

Updated At : Aug. 16, 2023, 2:44 p.m.

@copyright

Fig. 12.2.2 User Panel

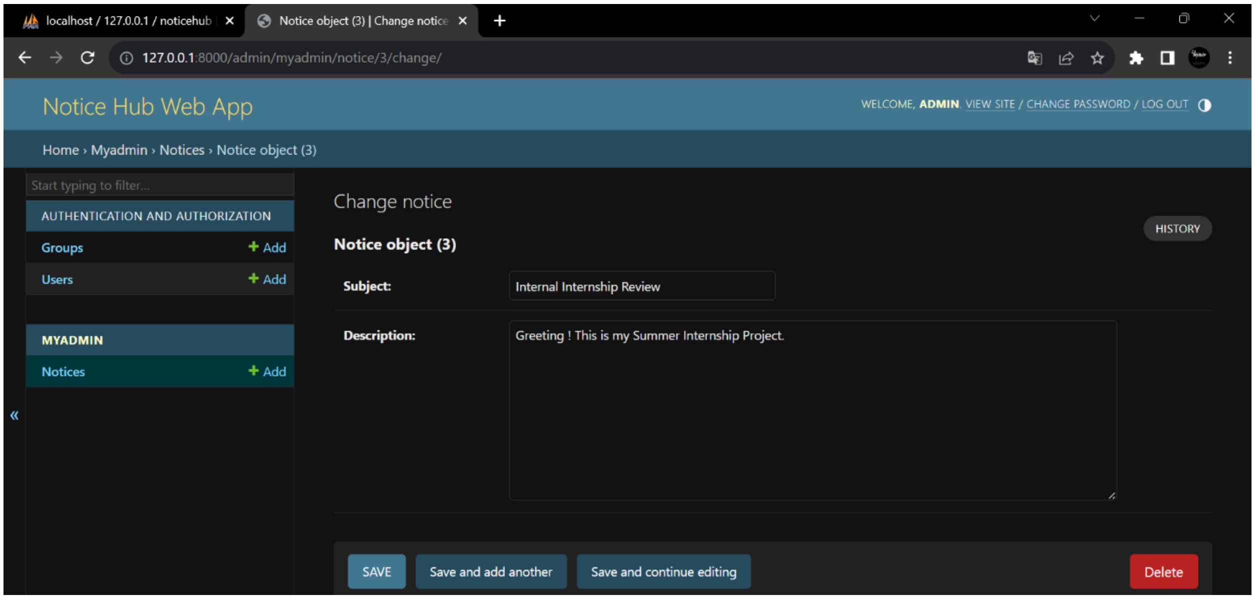

localhost / 127.0.0.1 / noticehub
Add notice | Django site admin
127.0.0.1:8000/admin/myadmin/notice/add/
Notice Hub Web App
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Myadmin › Notices › Add notice
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MYADMIN
Notices + Add
«
SAVE
Save and add another
Save and continue editing

Add notice

Subject:

Description:

Fig. 12.2.3 Create New Notice


localhost / 127.0.0.1 / noticehub
Notice object (3) | Change notice
127.0.0.1:8000/admin/myadmin/notice/3/change/
Notice Hub Web App
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Myadmin › Notices › Notice object (3)
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MYADMIN
Notices + Add
«
HISTORY
SAVE
Save and add another
Save and continue editing
Delete

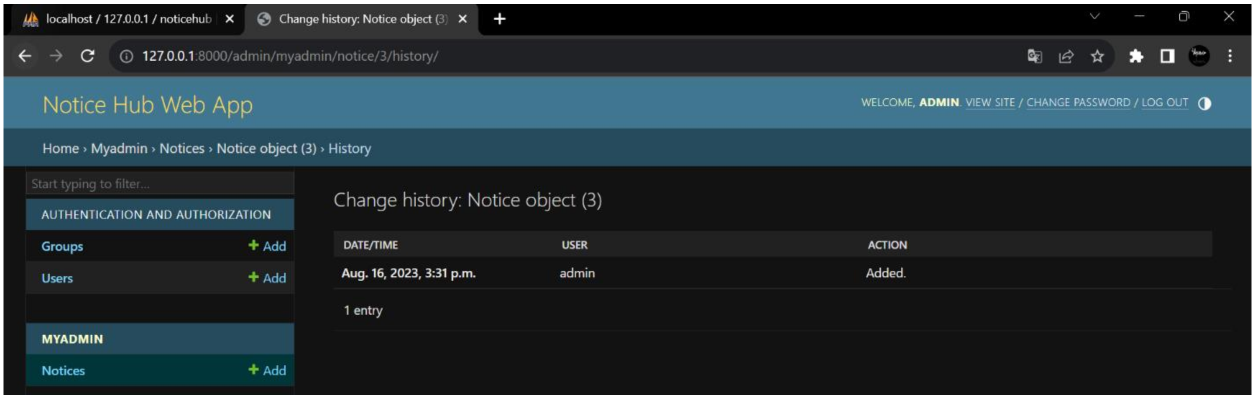
Change notice

Notice object (3)

Subject: Internal Internship Review

Description: Greeting ! This is my Summer Internship Project.

Fig. 12.2.4 Write Notice

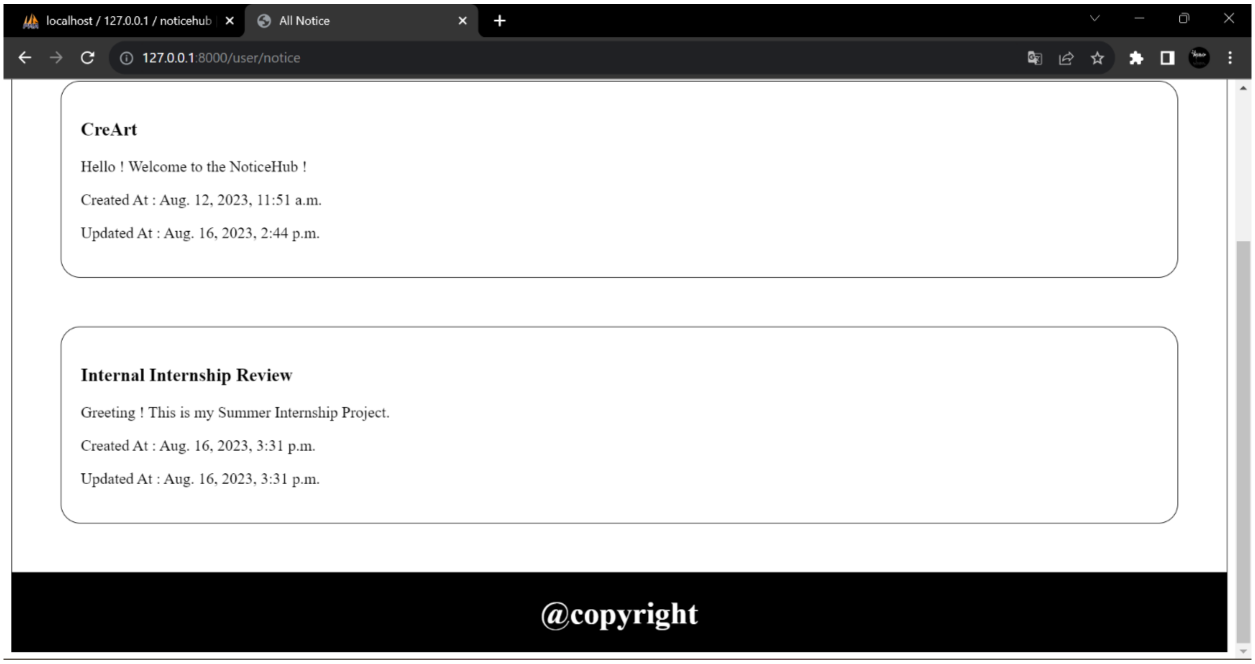

localhost / 127.0.0.1 / noticehub
Change history: Notice object (3)
127.0.0.1:8000/admin/myadmin/notice/3/history/
Notice Hub Web App
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Myadmin › Notices › Notice object (3) › History
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MYADMIN
Notices + Add

Change history: Notice object (3)

DATE/TIME	USER	ACTION
Aug. 16, 2023, 3:31 p.m.	admin	Added.

1 entry

Fig. 12.2.5 Submit Notice



CreArt

Hello ! Welcome to the NoticeHub !

Created At : Aug. 12, 2023, 11:51 a.m.

Updated At : Aug. 16, 2023, 2:44 p.m.

Internal Internship Review

Greeting ! This is my Summer Internship Project.

Created At : Aug. 16, 2023, 3:31 p.m.

Updated At : Aug. 16, 2023, 3:31 p.m.

@copyright

Fig. 12.2.6 Output of New Notice

* * *