NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name : Anandharaj R
Student ID :710321104301

College Name

Angel College of Engineering and technology

CAPSTONE PROJECT SHOWCASE

Project Title

Notes Sharing Web Application using Django Framework

Abstract | Problem Statement | Project Overview | Proposed Solution | Technology Used | Modelling & Results | Conclusion

Abstract

This project focuses on the development of a Notes Sharing Web Application using the Django framework. The application facilitates users to create, share, and manage their notes efficiently. The technology stack employed includes HTML, CSS, and JavaScript for the frontend, while Python, Django, and SQL are utilized for the backend functionality.

The frontend development involves crafting an intuitive user interface (UI) using HTML for structure, CSS for styling, and JavaScript for dynamic interactions. The UI allows users to seamlessly navigate through the application, create new notes, edit existing ones, and share them with other users.

On the backend, Django provides a robust framework for building scalable web applications.

Source :

Abstract

Python, being the primary language of Django, is used for implementing the core functionality of the application. Django's built-in features such as authentication, routing, and ORM (Object-Relational Mapping) streamline the development process.

SQL is employed as the database management system to store and manage the notes data efficiently. Through Django's ORM, SQL queries are seamlessly integrated into the application logic, enabling smooth interaction between the backend and the database.

Overall, this project aims to demonstrate the integration of frontend and backend technologies to develop a functional and user-friendly Notes Sharing Web Application using the Django framework.

1.Problem Statement

The problem statement for this project is to develop a collaborative notes sharing web application using the Django framework that addresses the following challenges:

1.Lack of real-time collaboration features in existing note-taking applications.
2. Absence of an intuitive and user-friendly interface for creating, editing, and organizing notes.
3. Insufficient customization options for note formatting and organization.
4. Security and privacy concerns when sharing sensitive information or collaborating on notes.
5. Inefficient data management and retrieval methods in existing solutions.

By addressing these challenges, the project aims to provide users with a comprehensive and secure platform for sharing notes and collaborating with others in real-time, thereby enhancing productivity and knowledge sharing in various domains.

Project Overview

Title: Notes Sharing Web Application using Django Framework

Objective: Develop a modern and intuitive web application for seamless notes sharing and real-time collaboration.

Features:

1. User-friendly Interface: Intuitive interface for creating, editing, and organizing notes.
2. Real-time Collaboration: Simultaneous editing and sharing of notes among multiple users.
3. Customization Options: Flexible formatting and organization features for personalized note-taking.
4. Secure Data Management: Implementation of robust security measures to protect user data and ensure privacy.

Project Overview

5. Django Framework: Utilization of Django framework for efficient backend development and database management.

Technology Stack:

Frontend: HTML, CSS, JavaScript
Backend: Python, Django, SQL

Outcome: A comprehensive notes sharing platform that enhances productivity and facilitates knowledge dissemination through seamless collaboration.

Proposed Solution

- Real-time Collaboration: Implement WebSocket technology for real-time communication between users, enabling simultaneous editing and sharing of notes.
- Intuitive User Interface: Design a modern and user-friendly interface using HTML, CSS, and JavaScript for easy note creation, editing, and organization.
- Customization Features: Provide rich formatting options and organizational tools to personalize note-taking experience.
- Robust Security Measures: Implement authentication, authorization, and encryption techniques to ensure data privacy and secure collaboration.

Source :

Proposed Solution

- Django Framework Integration: Utilize Django's powerful features for backend development, including ORM for database management and built-in security functionalities.

Outcome:

A collaborative notes sharing web application that offers seamless real-time collaboration, intuitive user experience, extensive customization options, and robust security measures to enhance productivity and knowledge sharing.

Source :

Technology Used

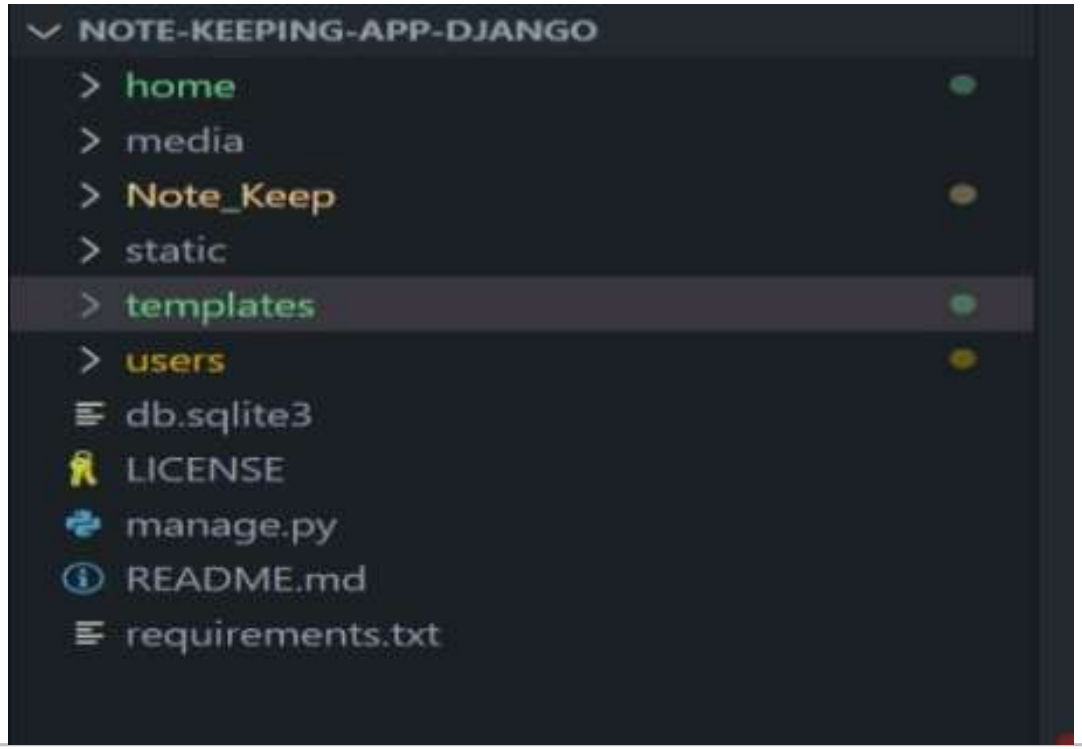
Front-end

Back-end

Source :

Modelling & Results

Folder Structure :



Source :

Modelling & Results

models.py :

home > models.py > ...

```python
from django.db import models

# Create your models here.
class Contact(models.Model):
    name = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    contact = models.CharField(max_length=10)
    desc = models.TextField()
    date = models.DateField()

    def __str__(self):
        return self.name
```

Source :

Modelling & Results

models.py :

```python
users > models.py > ...
from django.db import  models
from django.contrib.auth.models import User
from PIL import Image

# Create your models here.

class TODO(models.Model):
    status_choices = [
        ('C', 'COMPLETED'),
        ('P', 'PENDING')
    ]
    priority_choices = [
        ('1', '1'),
        ('2', '2'),
        ('3', '3'),
        ('4', '4'),
        ('5', '5')
    ]

    title = models.CharField(max_length=50)
    status = models.CharField(max_length=2, choices=status_choices)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    date = models.DateTimeField(auto_now_add=True)
    priority = models.CharField(max_length=5, choices=priority_choices)

    def __str__(self):
        return f'{self.user.username} ToDo'
```

Source :

Modelling & Results

models.py :

```python
class Note(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    title = models.CharField(max_length=150)
    content = models.TextField(blank=True,null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    modified_at = models.DateTimeField(auto_now=True)

    class Meta:
        ordering = ('title',)

    def __str__(self):
        return f'{self.user.username} Note'

class Profile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    image = models.ImageField(default='default.jpg', upload_to='profile_pics')

    def __str__(self):
        return f'{self.user.username} Profile'
```

Source :

Modelling & Results

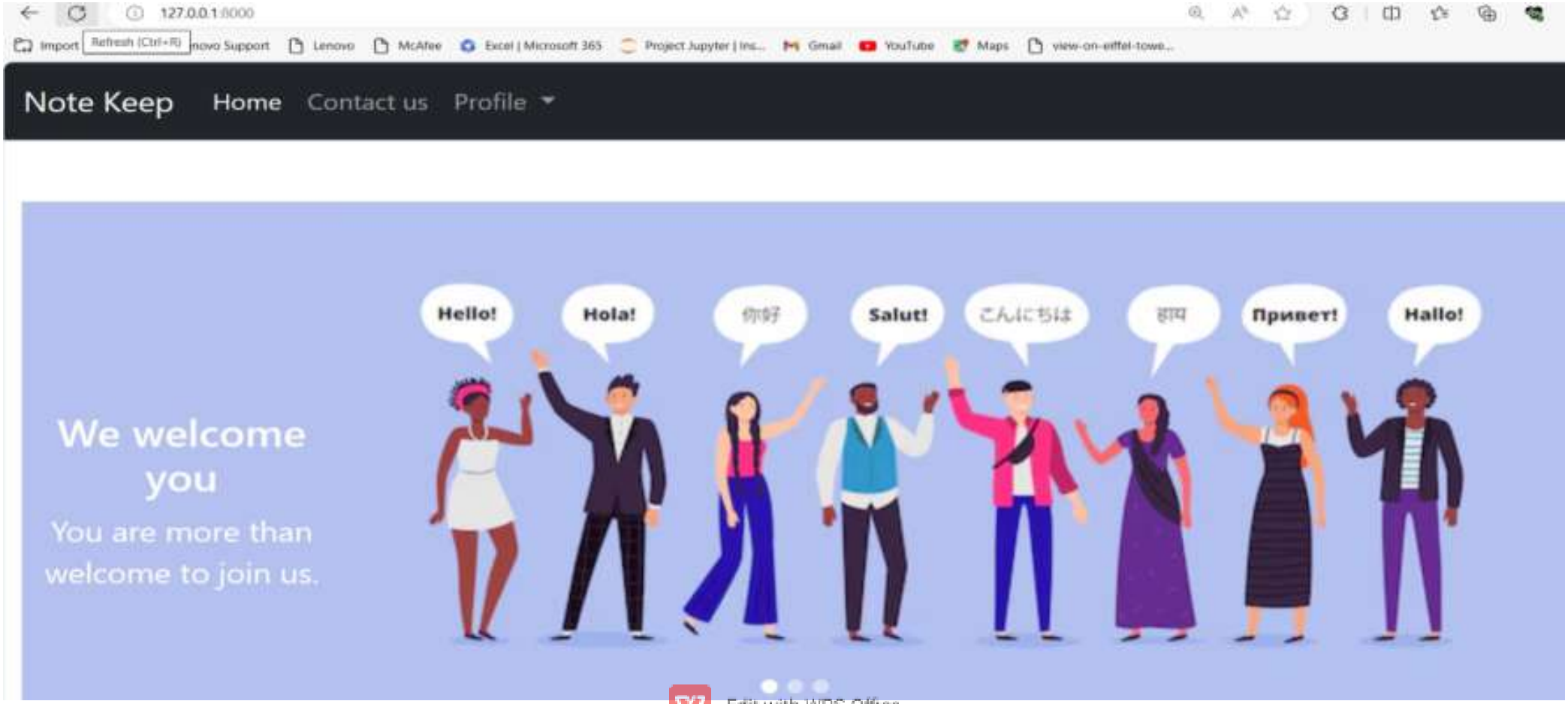
views.py :

home > views.py > ...

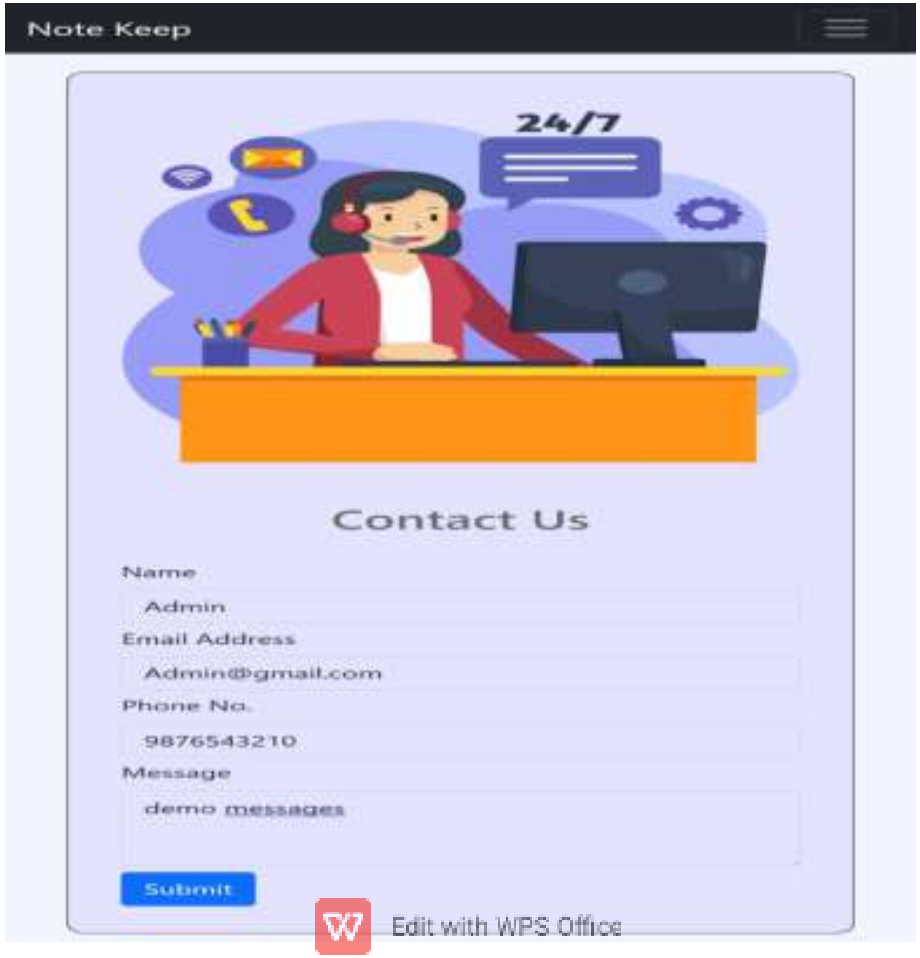
```python
from home.models import Contact
from django.shortcuts import render, HttpResponse
from datetime import datetime
from django.contrib import messages
# Create your views here.
def index(request):
    return render(request,"index.html")
    # return HttpResponse("Hello Django App")

def contact(request):
    if request.method == 'POST':
        name = request.POST.get('name')
        email = request.POST.get('email')
        contact = request.POST.get('contact')
        desc = request.POST.get('desc')
        contact = Contact(name=name, email=email, contact=contact, desc=desc, date=datetime.today())
        contact.save()
        messages.success(request, 'Your Form has been Submitted')
    return render(request,"contact.html")
```

Source :

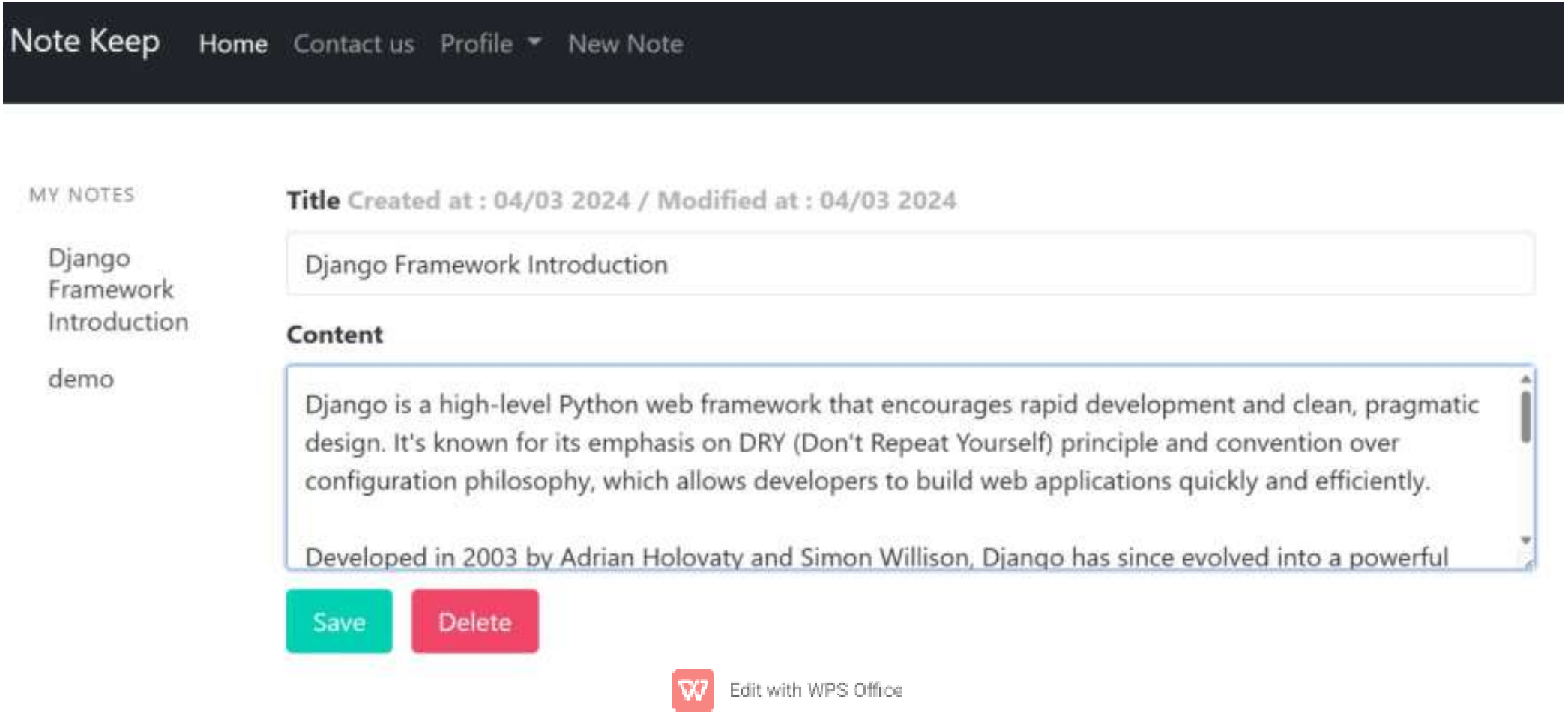
Homepage



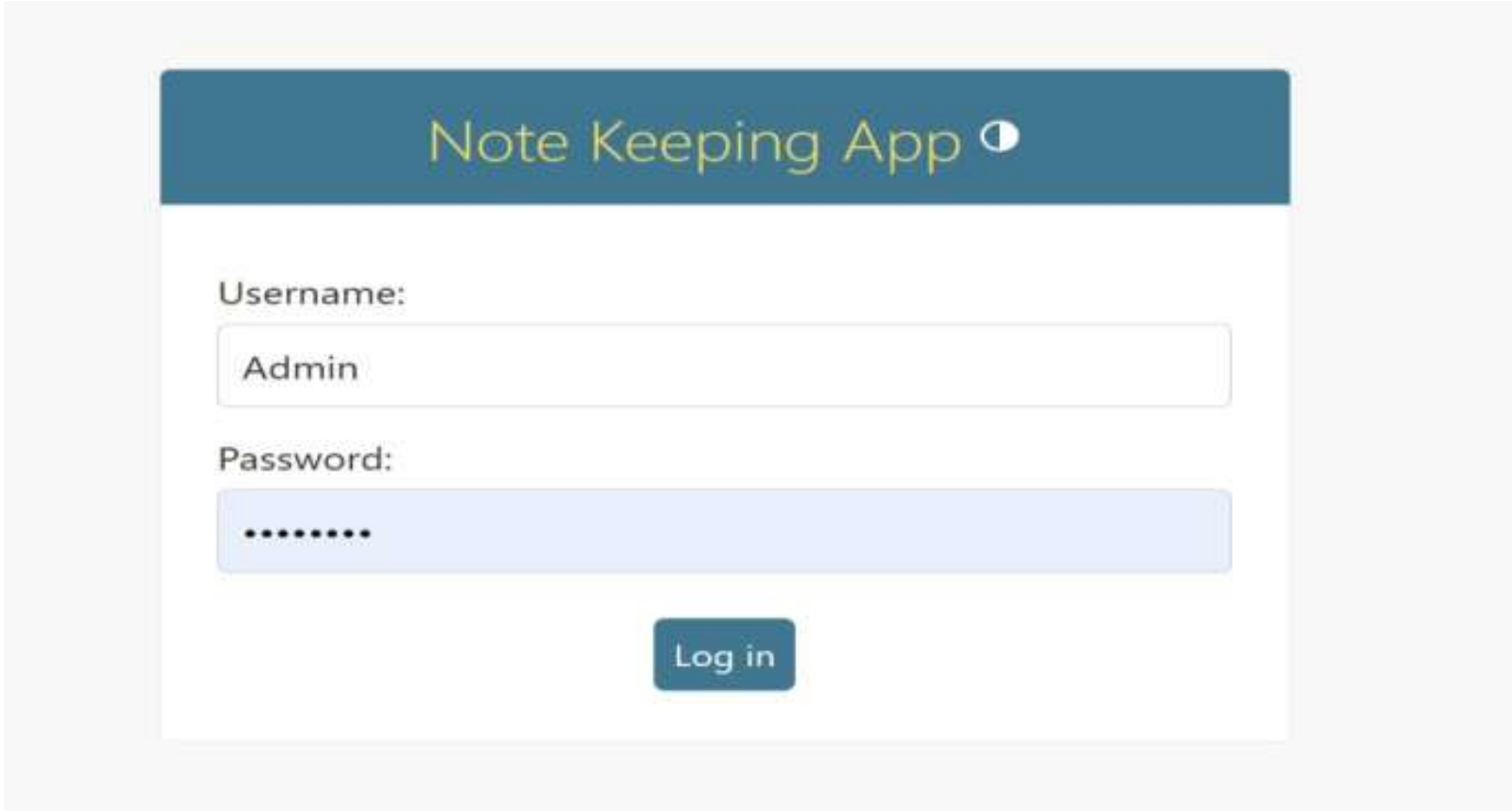
About-Us-Page



Service-Page

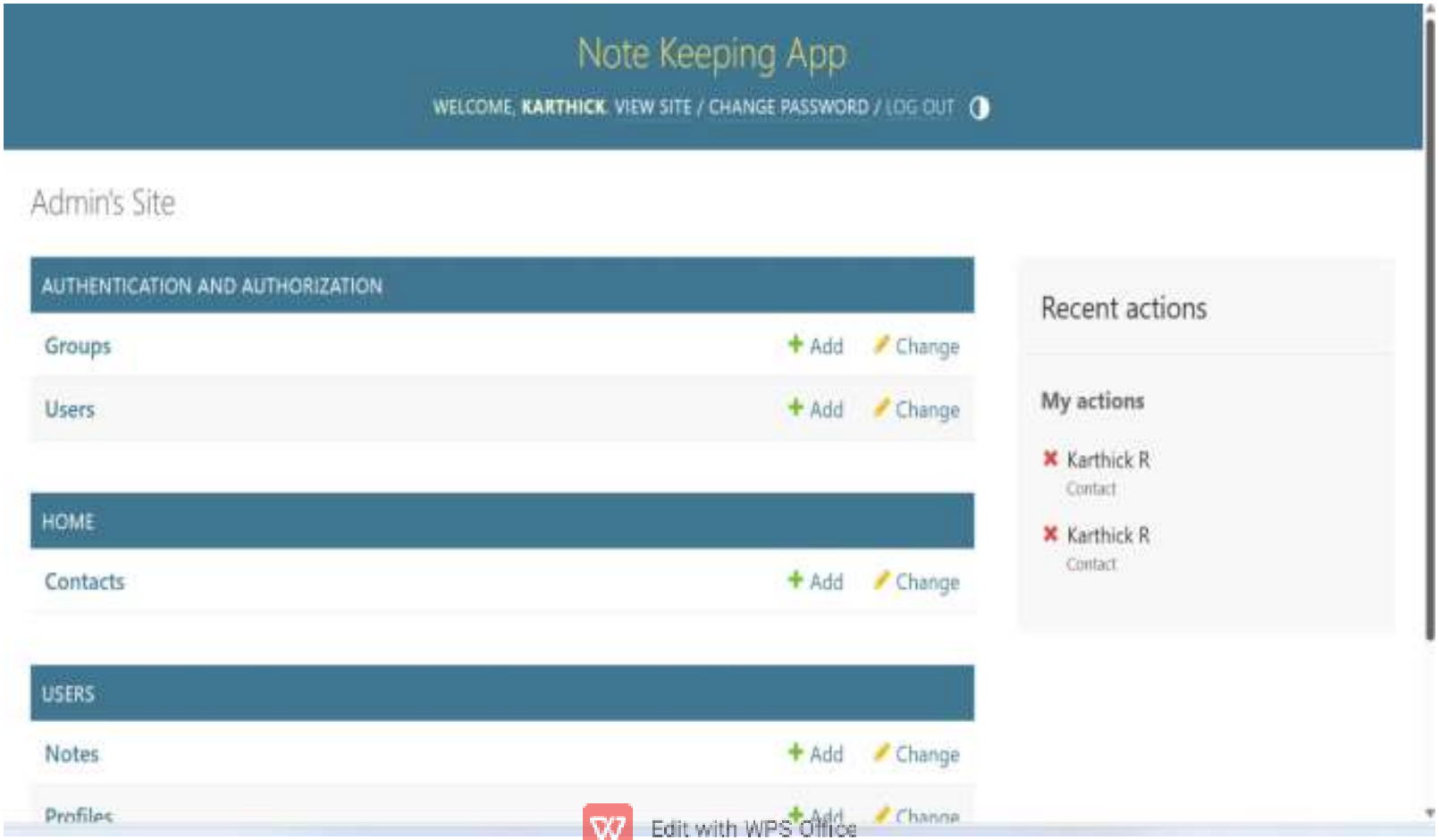


Login-Page

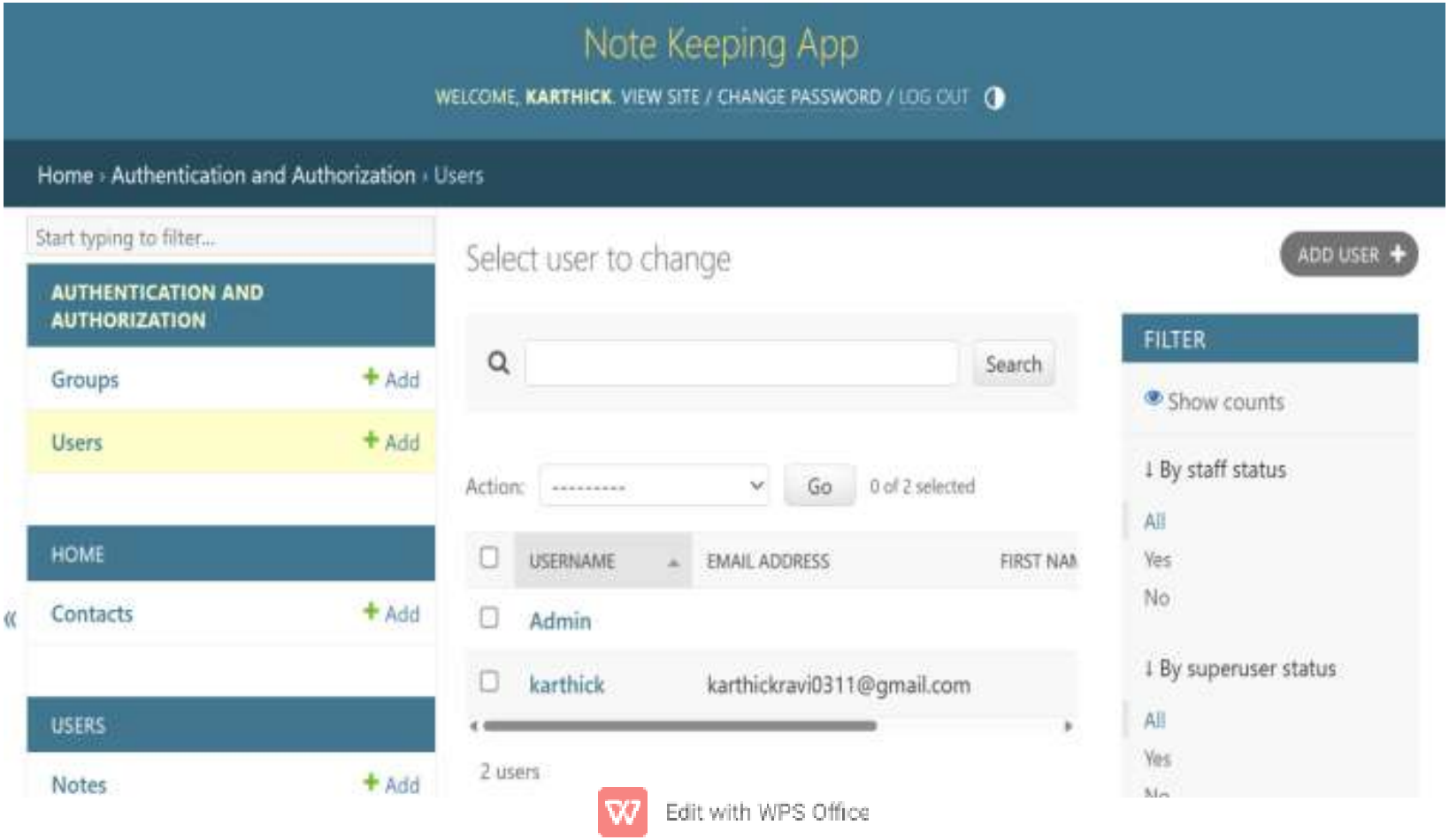


edunet foundation

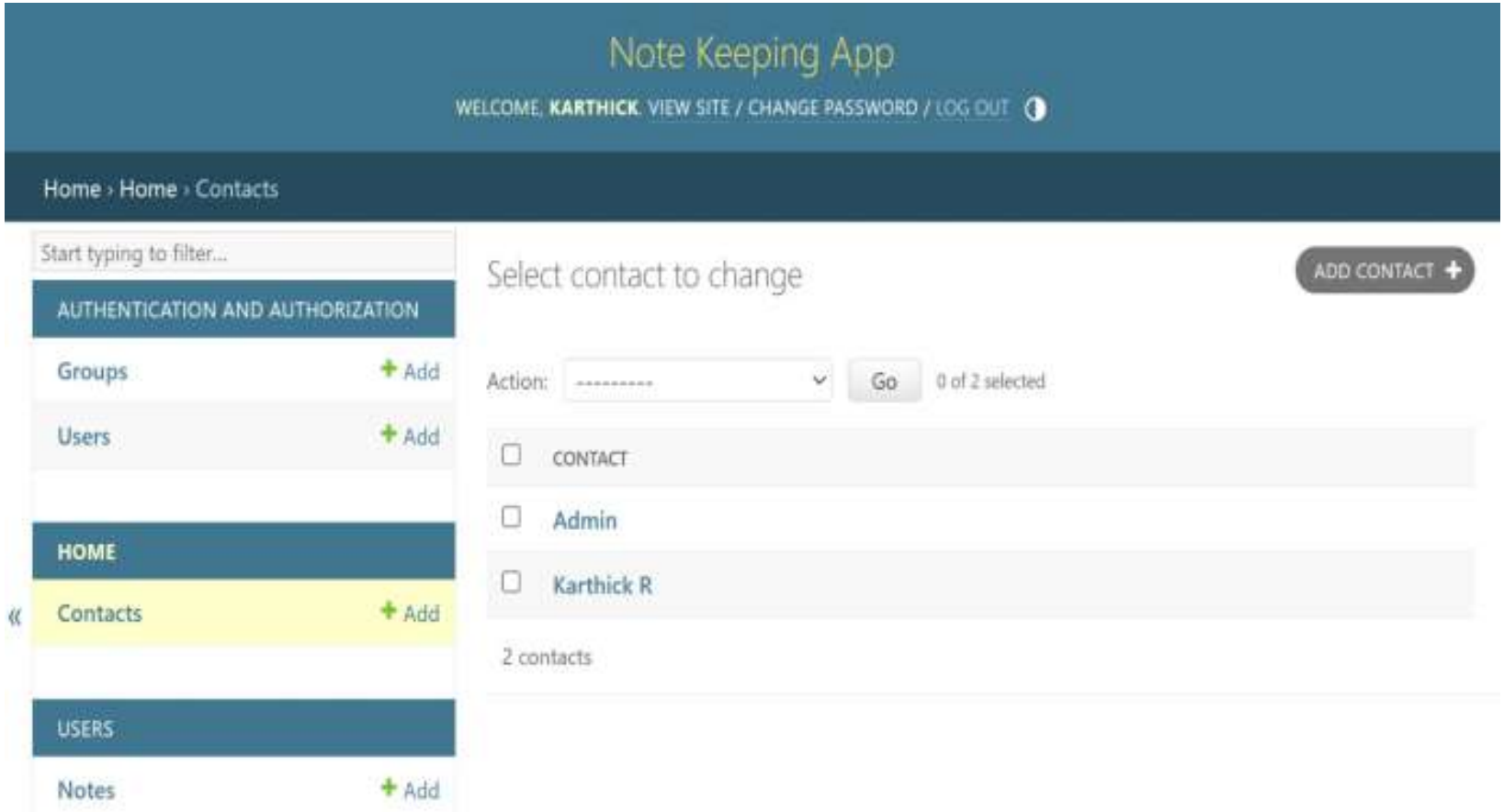
Admin-Page



Add/Update User-Page



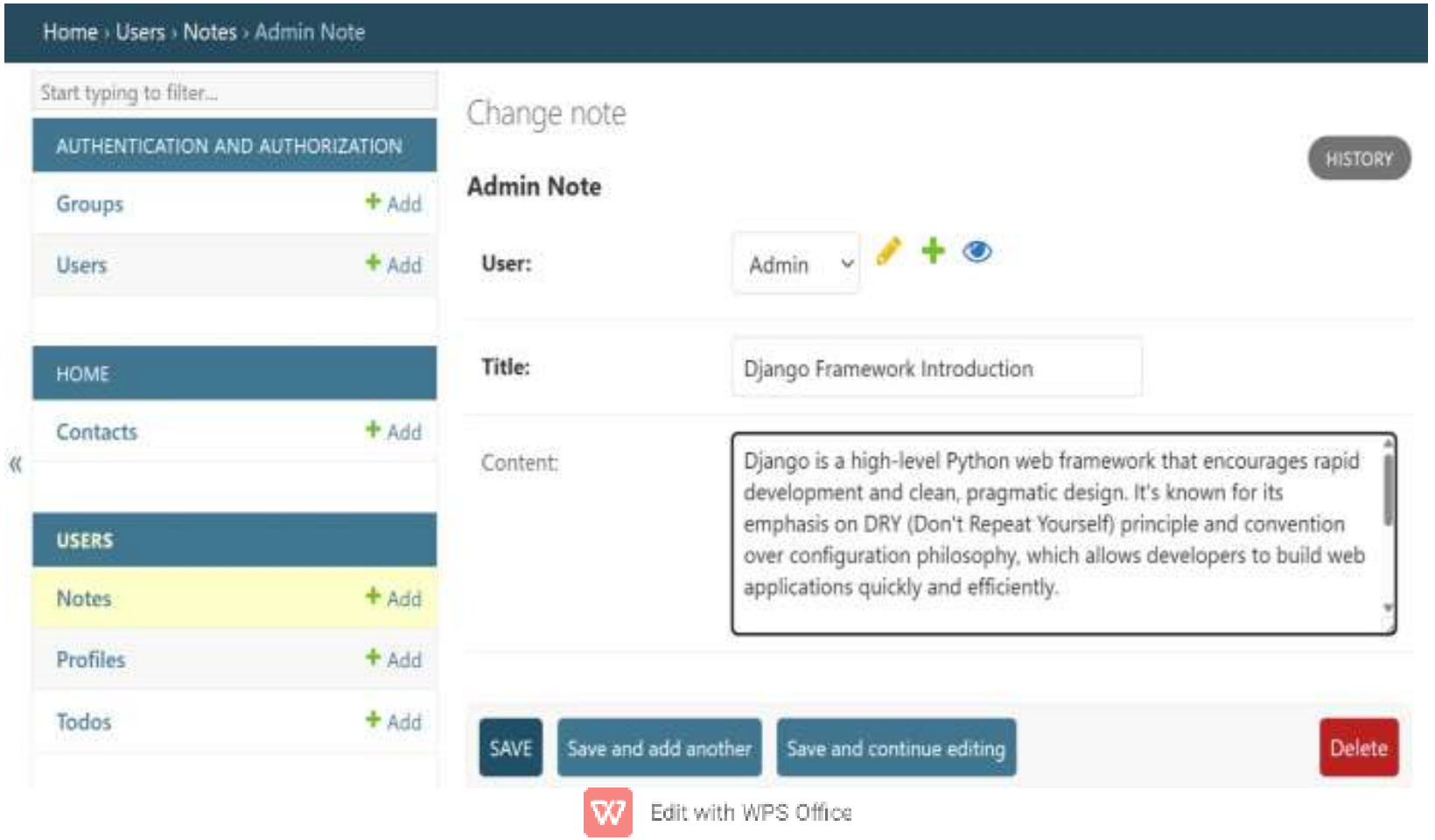
Add/Update Contact-Page



edunet foundation

Update profile-Page

Add Notes-Page



Future Enhancements

- Mobile Application: Develop a companion mobile app for the notes sharing platform to extend accessibility and convenience for users on-the-go.
- Enhanced Collaboration Tools: Integrate additional collaboration features such as real-time chat, version history tracking, and commenting to further facilitate teamwork and communication among users.
- Offline Mode: Implement offline functionality to allow users to access and edit notes even when they're not connected to the internet, with automatic synchronization once connectivity is restored.
- Advanced Search and Filtering: Enhance search capabilities with advanced filtering options, tag-based organization, and full-text search functionality to quickly locate and retrieve relevant notes.
- Multi-language Support: Add support for multiple languages to cater to a global user base and enhance accessibility for non-English speakers.

Conclusion

In conclusion, the development of a collaborative notes sharing web application using the Django framework presents a promising solution to address the challenges of real-time collaboration and seamless information sharing. By leveraging modern technologies such as HTML, CSS, JavaScript, Python, Django, and SQL, the project aims to provide users with an intuitive, secure, and feature-rich platform for managing and sharing their notes effectively.

Overall, the collaborative notes sharing web application represents a significant step forward in facilitating knowledge dissemination, enhancing productivity, and fostering collaboration among users. Through continuous iteration and improvement, the platform endeavors to meet the evolving needs of its users and remain a valuable tool for individuals and teams across various domains.

Source :

Thank You!