**Sistema de Monitoreo de Cultivos**

**VirtualConnection**

Una aplicación web diseñada para el monitoreo y gestión eficiente de cultivos agrícolas en tiempo real, ayudando a los agricultores a tomar decisiones informadas basadas en datos de temperatura del suelo y humedad

**Desarrollado por:** Daniel Camilo Gonzalez Funeme  
**Contacto:** camolo.777@gmail.com  
**Licencia:** MIT

Tabla de Contenidos

1…………………………Introducción

2…………………………. Objetivos del Proyecto

3…………………………. Descripción del Sistema

4…………………………. Instalación

5…………………………Configuración Adicional

6…………………………. Uso de la Aplicación

7…………………………. Tecnologías Utilizadas

* …………………-Python
* …………………- Django
* …………………- Firebase
* …………………-Charts.js
* …………………-ECharts.js

8…………………………Potencial de Crecimiento

9…………………………. Capturas

10…………………………. Contribución

11………………………. Licencia

**1. Introducción**

El **Sistema de Monitoreo de Cultivos** es una solución innovadora desarrollada para mejorar la gestión y supervisión de cultivos agrícolas. Esta aplicación web está construida con Python y el framework Django, proporcionando una herramienta eficaz para que los agricultores puedan monitorear sus cultivos de manera remota y en tiempo real. A través de la integración con Firebase, los usuarios pueden almacenar y visualizar datos cruciales como la temperatura del suelo y la humedad, permitiéndoles tomar decisiones informadas sobre el manejo de sus cultivos.

**2. Objetivos del Proyecto**

El proyecto tiene los siguientes objetivos principales:

1. **Monitoreo Eficaz:** Proveer una plataforma que permita a los agricultores monitorear el estado de sus cultivos en tiempo real.
2. **Gestión de Datos:** Facilitar el almacenamiento y acceso a los datos históricos de los cultivos, permitiendo analizar tendencias y patrones.
3. **Escalabilidad:** Diseñar una aplicación que pueda expandirse con nuevas funcionalidades, como riego automatizado y alertas personalizadas.

**3. Descripción del Sistema**

El **Sistema de Monitoreo de Cultivos** permite a los usuarios realizar las siguientes acciones:

* **Registro de Cultivos:** Los usuarios pueden registrar nuevos cultivos y asociar sensores para la monitorización del clima y el suelo.
* **Visualización de Datos:** Utilizando Chart.js, la aplicación ofrece gráficas que presentan datos históricos y en tiempo real de temperatura y humedad.
* **Alertas:** El sistema puede generar notificaciones para advertir sobre condiciones críticas, como temperaturas elevadas que podrían requerir riego adicional.

**4. Instalación**

#### Requisitos Mínimos del Sistema

Para asegurar un funcionamiento óptimo del sistema, es necesario que el equipo donde se ejecute cumpla con los siguientes requisitos mínimos:

* Procesador: AMD Ryzen 3 o Intel Core i3
* Memoria RAM: 2 GB
* Memoria de Video: 1 GB
* Espacio en Disco Duro: 100 GB de espacio libre
* Velocidad de Red: 10 Mbps
* Sistema Operativo: Windows 10, macOS, o una distribución moderna de Linux compatible con Python y Django

Para instalar el sistema, sigue estos pasos:

**1.** **Clonación del Repositorio:** Clona el repositorio del proyecto a tu máquina local.

git clone https://github.com/Dan2001afk/Proyecto-Etapa-Practica.git

cd Proyecto-Etapa-Practica

**2. Crear un Entorno Virtual**: Es recomendable usar un entorno virtual para manejar las dependencias del proyecto.

**En Linux/macOS:**

python3 -m venv venv

source venv/bin/activate

**En Windows:**

python -m venv venv

venv\Scripts\activate

**3. Instalación de Dependencias:** Instala Django y otras dependencias necesarias.

pip install -r requirements.txt

4. **Configuración de Firebase**

1. **Configura Firebase**:
   * Crea una cuenta en Firebase y habilita el servicio de Realtime Database o Firestore.
   * Descarga el archivo de credenciales JSON de Firebase.
2. **Coloca el archivo de credenciales**:
   * Coloca el archivo de credenciales descargado (virtualconnection.json) en la carpeta AppVirtualConnection del proyecto.
3. **Configuración en el Proyecto**:
   * En el archivo firebase\_config.py dentro de la carpeta AppVirtualConnection, asegúrate de tener la siguiente configuración:

import firebase\_admin

from firebase\_admin import credentials, firestore

# iniciamos la aplicacion de firebase con nuestras credenciales almacenadas en el archivo json

cred = credentials.Certificate("C:/Users/Daniel Gonzalez/Desktop/Proyecto-Etapa-Practica/VirtualConnection/AppVirtualConnection/virtualconnection.json")

if not firebase\_admin.\_apps:

firebase\_admin.initialize\_app(cred, {

'databaseURL': 'https://virtualconnection-643e6-default-rtdb.firebaseio.com/',

'storageBucket': 'virtualconnection-643e6.appspot.com'

})

db = firestore.client()

Asegúrate de cambiar la ruta "C:/Users/Daniel Gonzalez/Desktop/Proyecto-Etapa-Practica/VirtualConnection/AppVirtualConnection/virtualconnection.json" si es necesario, dependiendo de la ubicación del archivo en tu entorno.

**4. Base de Datos**:

* Asegúrate de configurar correctamente la URL de la base de datos y el bucket de almacenamiento en Firebase para que coincidan con los servicios habilitados en tu proyecto.

**5. Configuración Adicional: Almacenamiento de Archivos JSON**

En el archivo de vistas, se define una ruta que almacena los archivos JSON generados por la aplicación. Asegúrate de ajustar esta ruta según la ubicación en la que desees almacenar los archivos en tu entorno local.

1. En tu archivo de vistas (views.py), localiza la siguiente línea:

**RUTA\_GUARDAR\_JSON = os.path.join('C:/Users/Daniel Gonzalez/Desktop/Proyecto-Etapa-Practica/VirtualConnection/AppVirtualConnection/static/json')**

1. Cambia la ruta RUTA\_GUARDAR\_JSON a la ubicación donde quieras almacenar los archivos JSON en tu máquina o servidor. Por ejemplo, puedes definirla de esta manera:

**RUTA\_GUARDAR\_JSON = os.path.join('ruta/a/tu/directorio/json')**

1. Asegúrate de que el directorio exista y tenga permisos de escritura para que la aplicación pueda guardar los archivos JSON correctamente.

5. **Migraciones de Base de Datos:** Ejecuta las migraciones de Django para preparar la base de datos.

python manage.py migrate

**6. Ejecutar el Servidor:** Inicia el servidor de desarrollo para probar la aplicación localmente.

python manage.py runserver

**6. Uso de la Aplicación**

Una vez instalado, los usuarios pueden:

* **Registrar Cultivos:** Añadir nuevos cultivos al sistema, especificando detalles como el nombre del cultivo y los parámetros de monitoreo.
* **Visualizar Datos en Tiempo Real:** Acceder al dashboard para ver gráficos en tiempo real que muestran la evolución de las condiciones del cultivo.
* **Recibir Notificaciones:** Configurar alertas para recibir notificaciones cuando las condiciones del cultivo superen ciertos umbrales.

**7. Tecnologías Utilizadas**

**Python:** Lenguaje de programación utilizado para desarrollar la lógica del backend.

**Django:** Framework web que proporciona un desarrollo ágil y estructurado.

**Firebase:** Plataforma de backend que almacena datos de los cultivos y gestiona la autenticación.

**Chart.js:** Librería de JavaScript utilizada para crear gráficas dinámicas y visualmente atractivas

**ECharts.js**: una poderosa biblioteca de JavaScript de código abierto que se utiliza para crear gráficos interactivos y visualizaciones de datos. Desarrollada por Apache, es altamente personalizable y ofrece una amplia gama de gráficos, como:

* Gráficos de líneas
* Gráficos de barras
* Gráficos de dispersión
* Gráficos de torta
* Mapas geográficos
* Gráficos de radar
* Gráficos de embudo
* Gráficos personalizados

Algunas de sus características principales son:

1. **Interactividad**: ECharts permite agregar interacción a los gráficos, como hacer zoom, desplazarse o resaltar elementos al pasar el mouse.
2. **Altamente configurable**: Puedes personalizar cada aspecto del gráfico, desde colores hasta leyendas y estilos de las líneas.
3. **Gran cantidad de tipos de gráficos**: Soporta gráficos básicos como líneas o barras, y gráficos más complejos como mapas de calor o gráficos de árbol.
4. **Compatibilidad con grandes volúmenes de datos**: ECharts está optimizado para renderizar grandes conjuntos de datos de manera eficiente.
5. **Resposivo y adaptable**: Se ajusta automáticamente al tamaño de pantalla, lo que lo hace ideal para su uso en aplicaciones móviles y web.

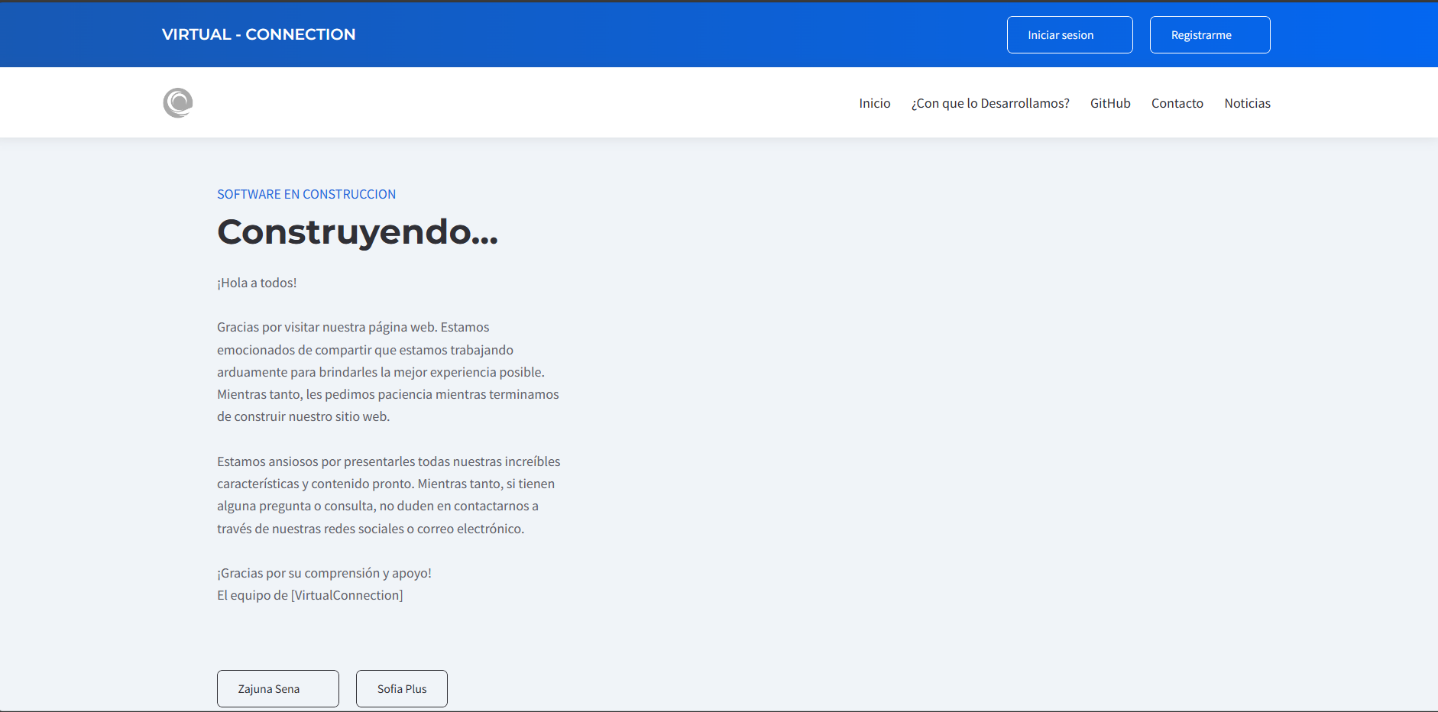
**8. Potencial de Crecimiento**

El **Sistema de Monitoreo de Cultivos** está diseñado para escalar y adaptarse a las necesidades futuras. Algunas mejoras planificadas incluyen:

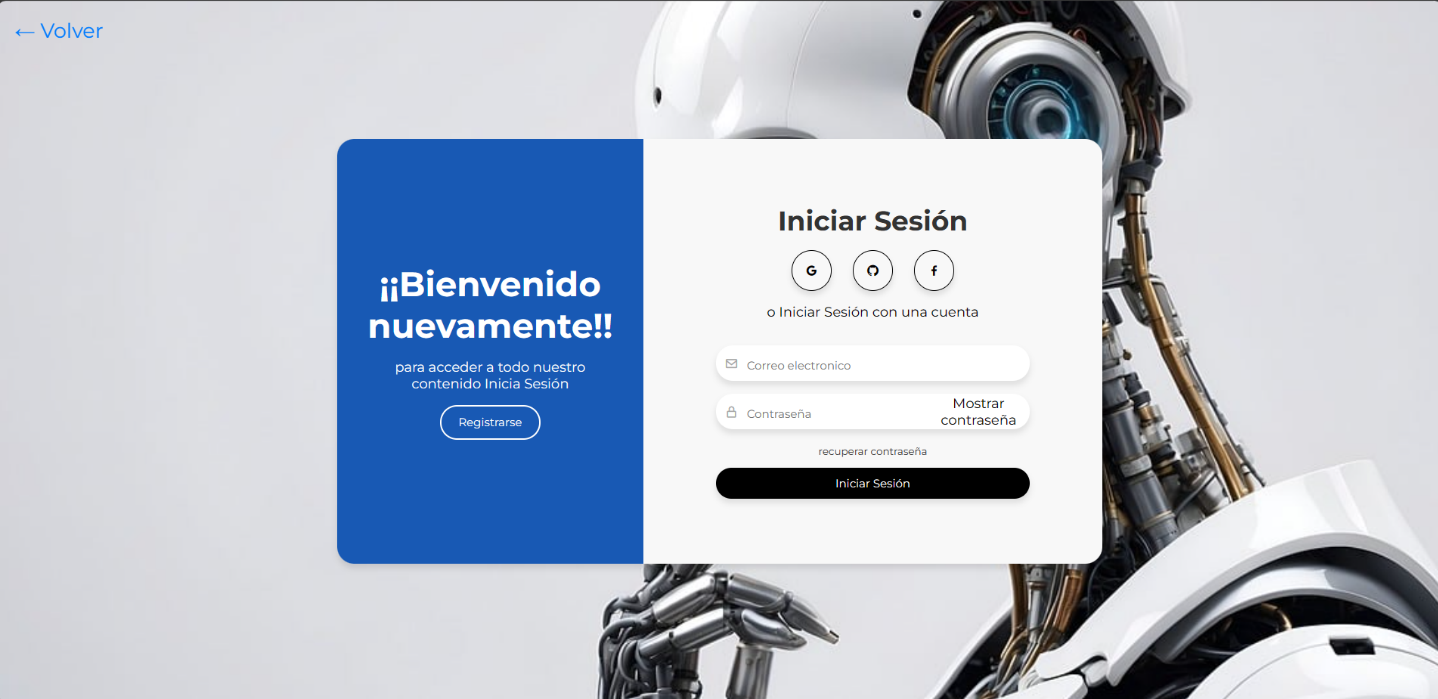
* **Automatización del Riego:** Integrar sistemas de riego automatizado que se activen según los datos de humedad del suelo.
* **Actualización en Tiempo Real:** Implementar tecnología de WebSockets o Firebase para actualizaciones instantáneas de los datos de cultivos.
* **Notificaciones Avanzadas:** Mejorar el sistema de notificaciones para alertar a los usuarios sobre condiciones meteorológicas adversas.

**9. Capturas**

**Inicio**

****

**Login**

****

**Registro**

**DashboardInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**10. Contribución**

¡Estamos abiertos a contribuciones! Si deseas contribuir al desarrollo de este proyecto, por favor sigue estos pasos:

1. Haz un fork de este repositorio.
2. Crea una rama nueva para tu contribución (**git checkout -b feature/nueva-caracteristica**).
3. Realiza tus cambios y commitea (**git commit -am 'Agrega nueva característica'**).
4. Sube tus cambios a tu repositorio fork (**git push origin feature/nueva-caracteristica**).
5. Crea una nueva solicitud de extracción en GitHub.

11. Licencia

MIT License

Copyright (c) [2024] [Daniel Camilo Gonzalez]

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.