Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

"Yo, como integrante de la comunidad estudiantil del Tecnológico de Monterrey, soy consciente de que la trampa y el engaño afectan mi dignidad como persona, mi aprendizaje y mi formación, por ello me comprometo a actuar honestamente, respetar y dar crédito al valor y esfuerzo con el que se elaboran las ideas propias, las de los compañeros y de los autores, así como asumir mi responsabilidad en la construcción de un ambiente de aprendizaje justo y confiable"

"Inteligencia artificial avanzada para la ciencia de datos I"

Evidencias del reto - Módulo 4

Interfaz

Equipo:

Frida Cano Falcón A01752953

Jorge Javier Sosa Briseño A01749489

Guillermo Romeo Cepeda Medina A01284015

Daniel Saldaña Rodríguez A00829752

Fecha de entrega: 11 de septiembre de 2023

Introducción

Siguiendo con la documentación de nuestra solución al reto que nos propuso nuestro socio formador, el cual involucra realizar una página web que lleve un rastro de asistencias y participaciones de alumnos en sus salones de clases. Es importante considerar que los datos que estamos manejando son confidenciales y pertenecen a la escuela en la que se utilizará esta plataforma de trackeo de asistencias. Estos datos son delicados pues incluyen la edad y el nombre,así como el trackeo de cuando y donde asisten a clases, información que en manos equivocadas puede poner en riesgo la seguridad de estas personas. Por lo que el proteger los datos en nuestra plataforma debe ser sumamente importante.

Por lo que se nos pide realizar lo siguiente:

1. Verificar que los datos que generes esten anonimizados, es decir que no se pueda rastrear información personal o sensible a una persona o producto específico a través del data set. Si los datos ya están anonimizados, describe cuáles fueron los atributos y las razones por las que se tienen que enmascarar.

La manera en la que registramos nuestros datos de forma anónima es a través de una encriptación con una llave aleatoria antes de llegar a nuestra base de datos, esta llave la utiliza el backend que se encuentra en el repositorio al que solo las personas autorizadas para desarrollar el proyecto tienen acceso, pero esto puede incluso volverse más seguro al tener esta llave aleatoria funcionando solo en el ambiente de producción, sin que los desarrolladores la tengan que ver, en nuestro caso solo encriptamos los nombres para simplificar el manejo de datos en la base, pero este concepto puede ser aplicado a cada uno de los datos que queramos encriptar para que sean anónimos para las personas en el desarrollo, pero que las personas que si los necesitan como los maestros y otras figuras los puedan ver en el frontend, esto se hace utilizando la misma clave pero en el frontend, así los datos nunca son visibles por los desarrolladores, aqui la prueba:

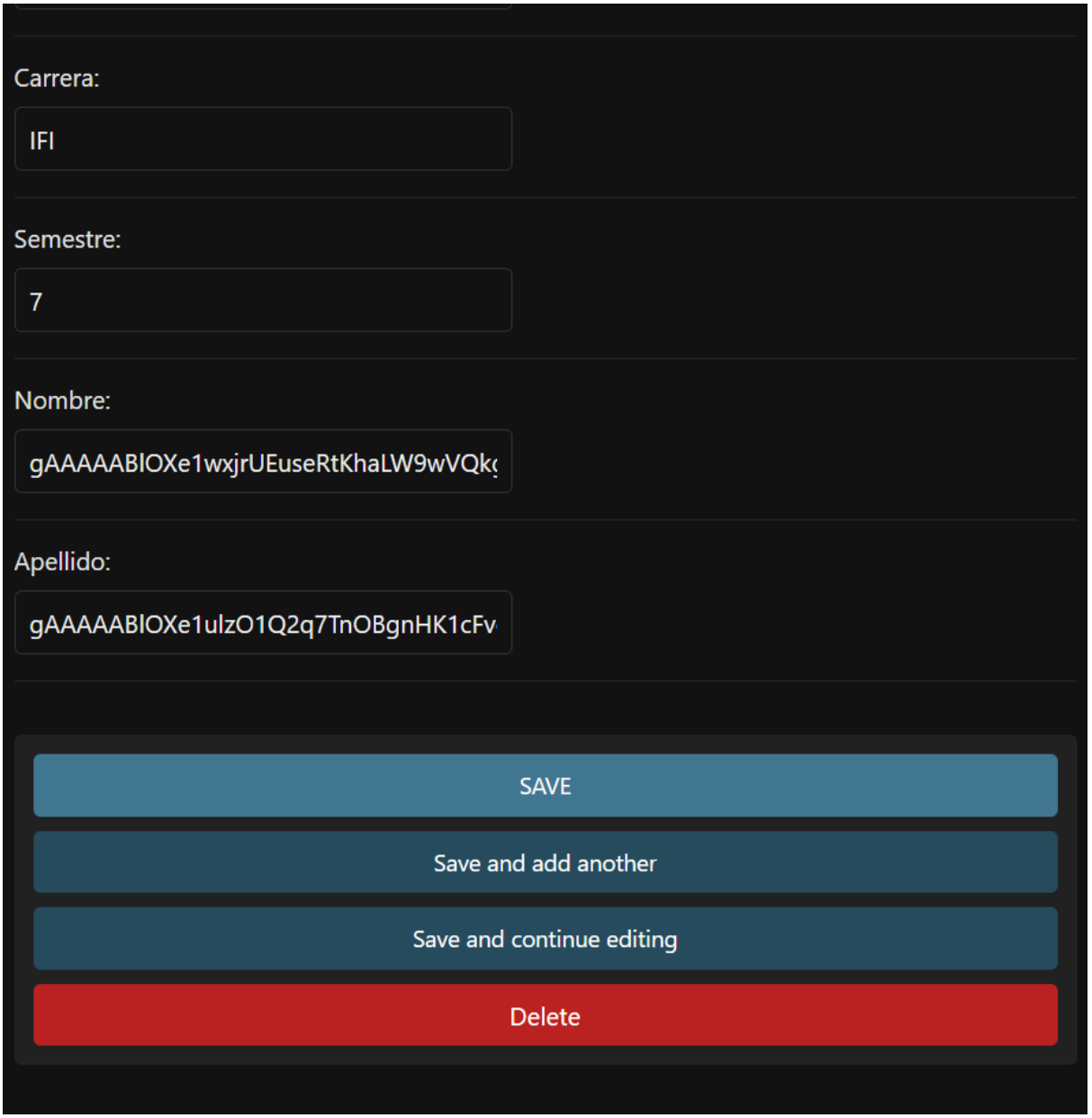
Encriptación en el backend:

```
from django.db import models

from django.contrib.auth.models import AbstractUser

from cryptography.fernet import Fernet



#utilizar la llave secreta para encriptar nombres de los usuarios

secret_key = b'HCSP-XM-YJ7L4Z_1HN7_Y86v75l0UZpejRBSq_CAv8A='

cipher_suite = Fernet(secret_key)



# Create your models here.
```

```
class CustomUser(AbstractUser):
    age = models.PositiveIntegerField(null=True, blank=True)
    matricula = models.CharField(max_length=10, null=True, blank=True)
    carrera = models.CharField(max_length=50, null=True, blank=True)
    semestre = models.CharField(max_length=10, null=True, blank=True)
    nombre = models.CharField(max_length=100, null=True, blank=True)
    apellido = models.CharField(max_length=100, null=True, blank=True)



    def encrypt_nombre(self, nombre):
        if nombre:
            encrypted_nombre = cipher_suite.encrypt(nombre.encode())
            #use only the string data
            encrypted_nombre = encrypted_nombre.decode()
            print(encrypted_nombre)
            return encrypted_nombre
        return None


    def save(self, *args, **kwargs):
        if self.nombre:
            self.nombre = self.encrypt_nombre(self.nombre)
        if self.apellido:
            self.apellido = self.encrypt_nombre(self.apellido)


        super().save(*args, **kwargs)


    def decrypt_nombre(self, encrypted_nombre):
        if encrypted_nombre:
                                                decrypted_nombre    =
cipher_suite.decrypt(encrypted_nombre).decode()
```

```
            return decrypted_nombre
        return None


    def get_nombre(self):
        return self.decrypt_nombre(self.nombre)
```

Visualización en el backend:



Sin embargo, en el front-end de los usuarios desplegamos los datos desencriptados:

```
from django.shortcuts import render, redirect



from .forms import AttendanceForm

# Create your views here.

from django.views.generic import ListView, DetailView

from .models import Attendance

from    django.views.generic.edit    import    CreateView,    UpdateView,
DeleteView

from django.urls import reverse_lazy

from cryptography.fernet import Fernet



def decrypt_name(encrypted_name):

    if encrypted_name:

        secret_key = b'HCSP-XM-YJ7L4Z_1HN7_Y86v75l0UZpejRBSq_CAv8A='

        cypher_suite = Fernet(secret_key)

        decrypted_name = cypher_suite.decrypt(encrypted_name).decode()

        return decrypted_name

    return None



class AttendanceListView(ListView):

    model = Attendance

    template_name = 'home.html'




    def get_context_data(self, **kwargs):



        context = super().get_context_data(**kwargs)

        # Decrypting nombre and apellido for each Attendance object

        for attendance in context['object_list']:
```

```
        #pass attendance.alumno.nombre to bytes
        fname = attendance.alumno.nombre.encode('utf-8')
        lname = attendance.alumno.apellido.encode('utf-8')
        attendance.alumno.nombre = decrypt_name(fname)
        attendance.alumno.apellido = decrypt_name(lname)
    return context
```

Visualización en el front-end

En nuestro caso enmascaramos los nombres porque es la información más sensible que tenemos actualmente en la base de datos, pero si se fueran a almacenar cosas como las direcciones de los hogares de los usuarios, datos bancarios, familiares, etc. También serán encriptados.

2. Consulta la normativa actual de la industria a la que esté sujeto el socio formador e investiga en reportes técnicos, artículos o foros cuales son los pasos comunes que se toman para garantizar la privacidad de los datos en dicha industria.

En el aviso de privacidad de NDS encontramos puntos importantes que nos hacen entender de mejor manera el papel de NDS en la protección de datos de sus usuarios.

Para empezar, notamos que el responsable en México es: NDS Cognitive Labs, en Prolongación Reforma 1190, Tower B, 21st floor, Colonia Cruz Manca, Cuajimalpa Delegation, C.P. 05349, in Mexico City, especifican que estos datos serán usados para identificación, operación, administración, y otros tratamientos para los servicios proveídos.

Ellos notifican al usuario de los derechos individuales de la siguiente ley: Federal Law on Protection of Personal Data Held by Private Parties (LFPDPPP), asi como Acces, Rectification, Cancellation and Opposition (ARCO rights). Esta ley es válida en México y regula el manejo y la protección de los datos personales por parte de las entidades privadas, esta ley establece que los datos personales sólo pueden ser recopilados y tratados por entidades privadas para fines específicos y legítimos e informados al titular de los datos, por lo que aquí cumplen en el aviso de publicidad. Asimismo la ley requiere el consentimiento del titular de los datos para ser utilizados. ARCO provee al usuario el derecho previamente mencionado a acceder, rectificar, cancelar u oponerse a sus datos.

Para cumplir estos requisitos de la ley, asi como otros, es necesaria la transparencia de datos así como el registro de dichas bases de datos ante las autoridades competentes, la ley asegura que se cumplan estas regulaciones al aplicar sanciones en los casos en los que se incumple con la LFPDPPP.

3. Establece un proceso claro sobre cómo se puede trabajar con el set de datos y especifica aspectos como: dónde se puede almacenar, en que tipo de redes puede estar, quien los puede ver y cuales son los documentos o normas que se deben de firmar antes de poder acceder a los datos.

Para trabajar con el set de datos se requieren las siguientes credenciales

Esquema -> Ejemplo: http://, https://, ftp://

Direccion IP -> Ejemplo (nombre de dominio): www.ejemplo.com o (dirección IP): 192.168.0.1

Puerto -> Ejemplo: 5000

Ruta: /ruta/ejemplo

Usuario y contraseña -> Ejemplo: Ejemplo: usuario:contraseña@.

Ejemplo completo: https://www.ejemplo.com:8080/ruta/de/ejemplo/?param1=valor1#seccion

Solo alguien con estos datos completos puede acceder a los datos, desde un workbench como MySQL workbench o DatGrip. Estos se pueden almacencar en un servicio en la nube como AWS, GoogleCloud, Microsoft Azure, etc o localmente en un servidor local. El documento

adecuado para el acceso de base de datos debe ser un NDA Non Disclosure Agreement, donde se acuerda la confidencialidad de los servidores y sanciones por romper dicho contrato.

4. Implementen un mecanismo o utiliza una herramienta que les permita establecer registros sobre quien y cuando tuvo acceso a los datos y bajo qué esquema. Estos registros los deberán integrar a su reporte como parte de la evidencia de final de módulo.

Agregar registro de acceso: En este primer paso, decidimos que era necesario agregar un sistema de registro cada vez que alguien accede a los datos. Esto implica capturar la información relevante, como el nombre de usuario, la fecha y la hora del acceso, así como el esquema bajo el cual se accede.

Base de datos de registros: Creamos una tabla en nuestra base de datos que utilizaremos para almacenar estos registros de acceso. La tabla contiene campos de usuario, fecha, hora y esquema para registrar toda la información necesaria.

Actualizar el código de acceso a los datos: Luego, procedimos a actualizar nuestro código de acceso a los datos de la aplicación para asegurarnos de que se agregue una llamada para registrar este acceso en la tabla de registros antes o después de que se realice el acceso, dondequiera que se realice el acceso a los datos. De esta manera, nos aseguramos de que estemos registrando cada acceso de manera automática y almacenando la información relevante en nuestra base de datos de registros.