НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Документация

към проект практика по програмиране и реализация на бази от данни, CSCB634
Специалност Информатика, 3 курс

Форум с автоматично модериране на груби коментари

Изготвили:
Атанасия Василева, F100360
Петър Тодоров, F105021
Дамир Рами, F114818
Максим Иванов, F107450
Христо Арабаджиев, F105103
Иво Навущанов, F104738

Съдържание

Проектът "Фофум с автоматично засичане на груби коментари" има за цел да създаде онлайн форум, който автоматично анализира коментарите на потребителите за наличие на негативни емоции и грубо съдържание, използвайки машинно обучение (ML.NET 2.0) и алгоритъм NAS-BERT. Форумът е реализиран с .NET Core и предоставя различни функционалности за потребители, модератори и администратори

- **Основни Функционалности**

1. Регистрация и управление на потребители: потребителите целенасочено са ограничени до възможността единствено да създават акаунти, а администраторите могат да активират и деактивират потребителски профили.

2. Публикуване и анализ на коментари: потребителите могат да публикуват коментари, които автоматично се анализират за наличие на негативни емоции чрез предварително обучен модел.

3. Модерация на коментари: коментари, маркирани като съдържащи грубо съдържание, се подават на модератори за преглед и одобрение или отхвърляне.

4. Управление на роли и права: администраторите управляват правата на модераторите и могат да активират или деактивират потребителски профили

Технологии и инструменти

1. .NET Core: за разработка на уеб приложението.

2. ML.NET 2.0: за имплементация на модела за анализ на коментарите.

3.NAS-BERT: алгоритъмът, използван за анализ на текстовете.

4. Entity Framework Core: за управление на базата данни.

5. ASP.NET Core Identity: За управление на потребители и роли

Архитектура на системата

Слоеве и компоненти:

1.Представителен слой (Presentation Layer):

Контролери (Controllers): обработват HTTP заявки и връщат отговори.
Изгледи (Views): отговорни за визуализацията на данните.

2.Бизнес слой (Business Layer):

Сервиси (Services): имплементират бизнес логиката и взаимодействат с базата данни.

3.Данни и достъп до данни (Data Access Layer):

Контекст на базата данни (DbContext): Управлява достъпа до базата данни.
Модели (Models): Определят структурата на данните.

1.Controllers <--> Services: контролерите използват услугите, за да изпълняват бизнес логиката.

2.Views <--> Controllers: изгледите взаимодействат с контролерите за показване на данни и приемане на потребителски вход.

3.Services <--> DbContext: услугите използват контекста на базата данни за извличане и записване на данни.

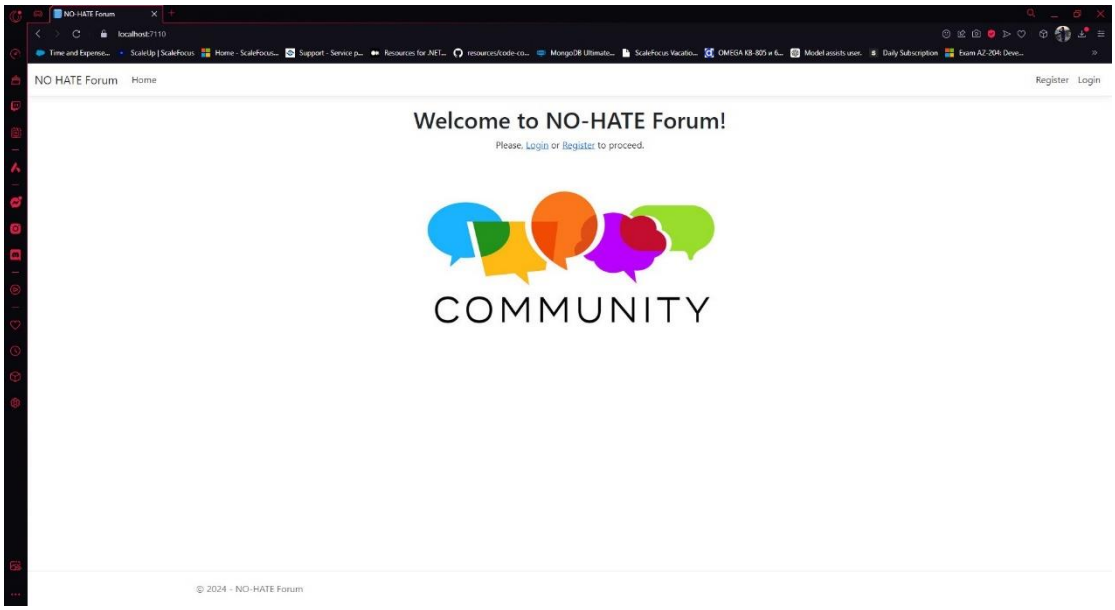
4.DbContext <--> Database: контекстът на базата данни осъществява връзката с физическата база данни.

5. Models <--> DbContext: моделите се използват от контекста на базата данни за работа със структурираните данни

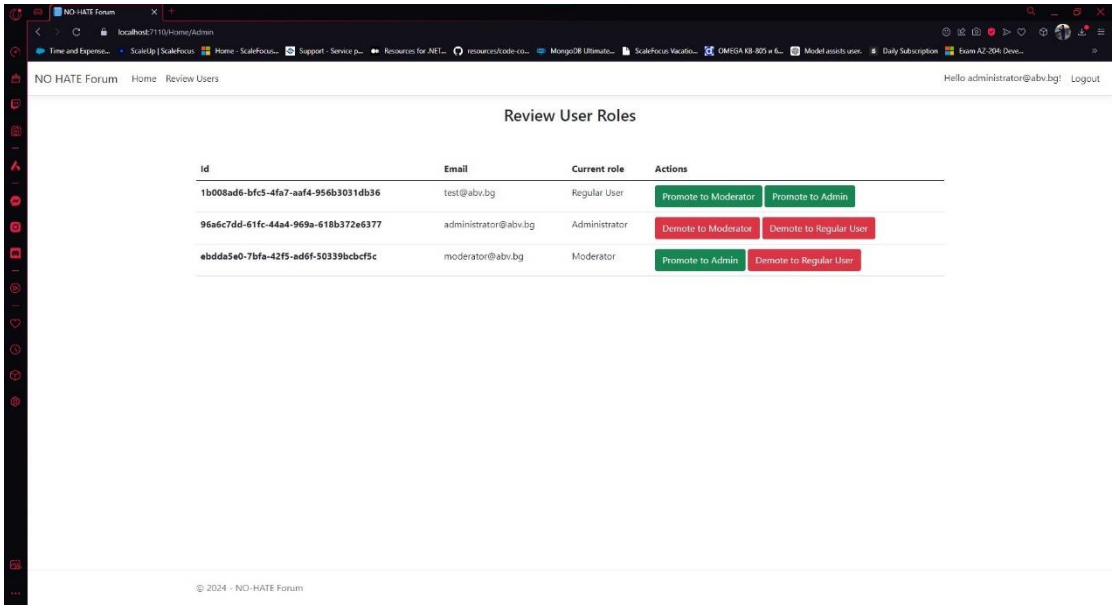
Реализация на проекта

Регистрация и управление на потребители

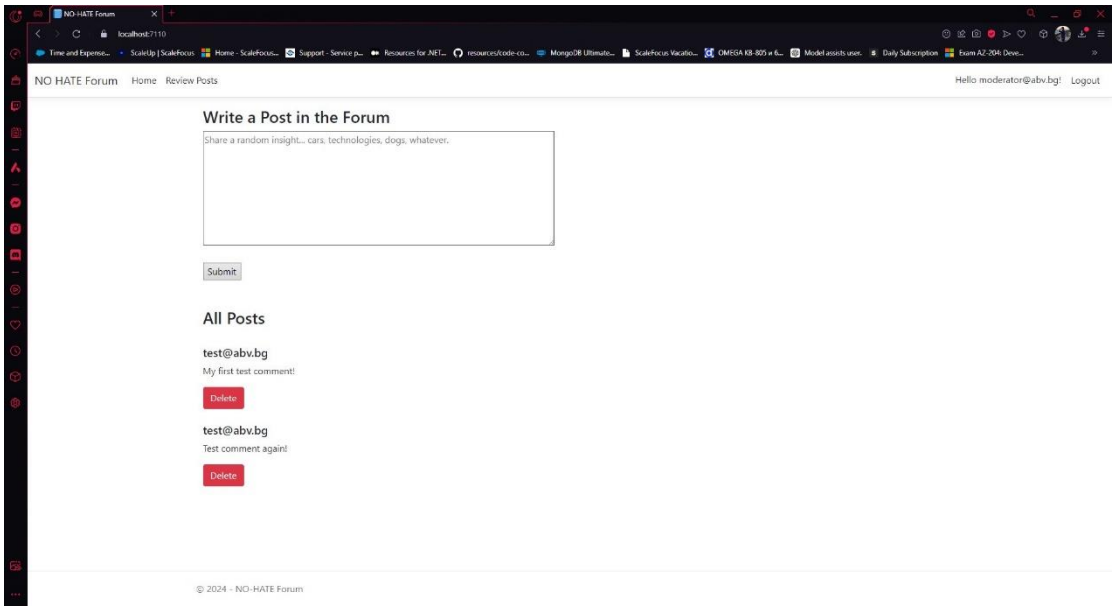
Потребителите могат да създават акаунти чрез предоставяне на потребителско име и парола. Администраторите имат възможност да активират и деактивират потребителски профили, както и да задават роли на потребителите (потребител, модератор, администратор).



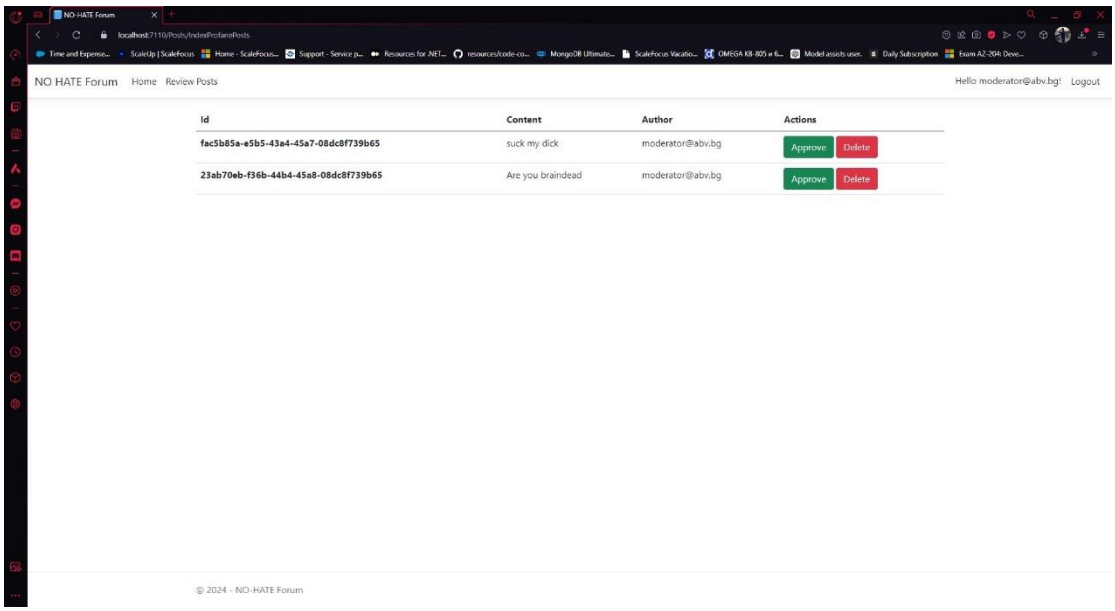
Guest page



Admin page



Home page



Moderator page

Решихме да заложим на изчистен минималистичен дизайн, който да е лесен за разбиране и удобен за ползване от всички потребители, без значение от техните роли и правомощия във форума.

На заглавната страница, или страницата за гости, присъства лого и опции за вписване и регистрация към форума.

На последващ етап, форумът ни отвежда към поле за въвеждане на текст, където да споделим текст, който в последствие бива автоматично проверен за груби коментари. На същия екран се виждат и вече споделените постове от други потребители и техните имена във форума.

Другите две ключови роли са на модератора и администратора. Налични са и специални страници за тях във форума, където могат да виждат всичко, което се случва и да контролират и управляват постовете във форума. На страницата на администратора се вижда, че той може да определя кой каква роля да изпълнява, както и всичката информация за хората, присъстващи във форума. На страницата на модератора са прихванати коментари с грубо съдържание и се очаква последваща реакция от страна на модератора. Той има опцията да одобри или изтрие коментарите, като може да види от кого точно е написан поста и какво е съдържанието, както и неговото ID.

Публикуване и анализ на коментари

Потребителите могат да публикуват коментари, които се анализират автоматично за наличие на негативни емоции и грубо съдържание. Този анализ се извършва чрез предварително обучен модел, използващ алгоритъма NAS-BERT. Коментарите, маркирани като негативни, се подават на модератори за преглед.

Модерация на коментари

Коментари, които са маркирани като съдържащи грубо съдържание, се подават на модератори за преглед. Модераторите имат възможност да одобрят или отхвърлят тези коментари. Ако коментарът бъде одобрен, той се показва публично във форума.

Управление на роли и права

Администраторите имат възможност да управляват правата на модераторите, както и да активират или деактивират потребителски профили. Това управление на роли и права се извършва чрез административния панел на форума

Подробно описание на компонентите

1.**Модели** – моделите представляват структурата на данните в системата. Основните модели включват:

- User: Представлява потребител в системата, който включва свойства като ID, Username, Password, IsActive и Role.

- Post: Представлява коментар, публикуван във форума, и включва свойства като ID, Content, AuthorId, Author и IsFlagged.

Контекст на базата данни - контекстът на базата данни (ApplicationDbContext) управлява връзката с базата данни и осигурява достъп до моделите чрез DbSet<T> свойства. Той също така конфигурира връзката към базата данни с помощта на Entity Framework Core.

2.Сървиси - PostService отговаря за бизнес логиката, свързана с управлението на коментарите. Основни методи включват:

GetAll: Извлича всички коментари от базата данни.

GetProfanePosts: Извлича всички коментари, маркирани като съдържащи грубо съдържание.

Create: Създава нов коментар и проверява дали съдържа груб език с помощта на PredictionEngine.

Delete: Изтрива коментар от базата данни.

ContainsProfanity: Проверява дали съдържанието на коментара съдържа груб език, използвайки PredictionEngine.

Контролери

- HomeController отговаря за обработка на заявки към началната страница на форума. Той извлича и показва всички коментари, като предоставя различни изгледи за автентицирани и неавтентицирани потребители.

- PostsController отговаря за обработка на заявки, свързани с публикуването и модерирането на коментари. Основни методи включват:

- Index: Извлича и показва всички коментари.

- IndexProfanePosts: Извлича и показва всички коментари, маркирани като съдържащи грубо съдържание.

- Create: Публикува нов коментар след проверка за груб език.

- Delete: Изтрива коментар.

- UsersController отговаря за управлението на потребителите в системата. *Основни методи включват:*

- CreateUser: Създава нов потребител.

- ActivateUser: Активира потребителски профил.

- DeactivateUser: Деактивира потребителски профил.

- UpdateUserRole: Актуализира ролята на потребител.

Конфигурация и инициализация

Startup.cs - файлът Startup.cs конфигурира услугите и средата на приложението. Той включва конфигурация на базата данни, идентичността, автентикацията и авторизацията, както и настройките за маршрутизация и управление на HTTP заявки.

ConfigureServices: Методът регистрира всички необходими услуги, като базата данни, идентичността, PredictionEnginePool и контролерите.

Configure: Методът конфигурира HTTP заявките, маршрутизацията, автентикацията и авторизацията.

Конфигурационни файлове:

appsettings.json - конфигурационният файл appsettings.json съдържа настройки за връзката към базата данни и логването.

Други компоненти:

ModelInput и ModelOutput - тези класове представляват входните и изходните данни за модела за машинно обучение.

ModelInput: Съдържа свойството Content, което представлява текстът на коментара.

ModelOutput: Съдържа свойството IsProfane, което показва дали текстът съдържа груб език.

ProfanityData и ProfanityDetectionTrainer - Тези компоненти са свързани с обучението на модела за машинно обучение.

ProfanityData: Класът, който представлява данните за обучение с текст и етикет (показващ дали текстът е профанен).

ProfanityDetectionTrainer: Класът, който управлява процеса на обучение на модела и неговата оценка

Проектът "Non-HateSpeech-Forum" предоставя сигурна и контролирана среда за комуникация, като използва машинно обучение за автоматичен анализ на коментарите и модерация на съдържанието. Чрез ясна структура и използване на съвременни технологии, форумът осигурява ефективна платформа за управление на онлайн комуникациите.