9Chan Diskussionsforum

Avlämningsrapport

Kravspecifikation av beställt beställd webbsida och beskrivning

* Admin ska kunna skapa kategorier samt underkategorier till kategorien
* Vem som helst ska kunna registrera sig och kunna ta del av innehåll
* En registrerad användare ska kunna ladda upp en profilbild som visas vid användarens inlägg på sidan
* Vem som helst ska kunna kommentera inlägg
* Inlägg ska ha ett datum då det skapades
* Skicka privata meddelanden till användare på hemsidan
* Anmälan av oönskade inlägg
* Lista över oönskade inlägg som admin kan ta bort eller av rapportera inlägg
* Api för att kunna hemsta inlägg ur en tråd med datum och författare
* Inlägg ska arrangeras i trädliknande struktur, där enskilda inlägg kan kommenteras så att en sub diskussion utifrån inlägget skapas
* Inlägg ska kunna ha bilder i sig
* Grupper med gruppmeddelanden
* Gilla funktion av inlägg
* Filter av svordomar
* Egen sida där användaren kan göra en presentation av sig själv

Extra funktioner

* Meddelanden sparas krypterat
* Email api för konto verifikation
* Se datum och vem som skapat senaste inlägg samt hur många inlägg finns i tråd
* Admin kan skapa “Sticky” tråd som hamnar längst upp i tråd listan
* Se datum och vem som skapat senaste tråden samt hur många trådar det finns

Projektet började 2021-05-17 och avslutades 2021-06-15

## Tekniska specifikationer

* .Net 5.0
* Azure Sql Server
* Azure Web Services
* Azure Secret Vault
* Azure Sql Database

### Nuget Packages

Azure.Extensions.AspNetCore.Configuration.Secrets" Version="1.1.0”

Azure.Identity" Version="1.4.0"

Microsoft.AspNetCore.Identity.EntityFrameworkCore" Version="5.0.6

Microsoft.AspNetCore.Identity.UI" Version="5.0.6"

Microsoft.EntityFrameworkCore.SqlServer" Version="5.0.6"

Microsoft.EntityFrameworkCore.Tools"

Microsoft.Extensions.Configuration.AzureKeyVault"

Microsoft.VisualStudio.Web.CodeGeneration.Design"

SendGrid" Version="9.23.2"

## Applications Struktur

ForumSite.csproj där information presenteras

9Chan.Data.csproj hanteras databaser och Api:er

9Chan.Core.csproj databas modeler samt Option classer för EmailSender och Encryption Service

ProfanityFilter.sln (fula ord microservice)

## Beroenden

* CategoriData ICategoryData
* SubcategoryData ISubCategoryData
* ThreadData IThreadData
* PostData IPostData
* CommentData ICommentData
* MessageData IMessageData
* LikeData ILikeData
* GroupData ILikeData
* UserGroupManager IUserGroupManager
* ProfanityFilter IProfanityFilter
* EmailSender IEmailSender
* AuthMessageOptions
* Encryption IEncryption
* EncryptionOptions

Sidor som behöver tillgång till tex Sparad data använder sig ut av respektive Interface genom dependency injection tex sidan Categories behöver data från Categories tabellen i databasen samt subcategories tabellen för att visa det som behövs på sidan.

AuthMessageOptions till email sendern hämtar sina värden i Azure Vault secrets för att kunna authentisiera sig till email sender apiet, Encryption fungerar på samma sätt.

## Installation

* I Mappen Areas/Identity/IdentityHostingStartup.cs av kommentera den databas du vill använda Azure eller lokal databas
* Använd Package Manager Console och välj \_9Chan.Data och ange update Database
* I Startup.cs kommentera ut Admin Policy
* Starta upp applikationen och klicka på Admin för att komma till admin sidan.
* Gå in på roller och skapa 2 roller “Admin” och “User”
* Registrera din användare du vill ha som Admin
* Gå till Admin sidan och in på roller och tryck på Promote knappen vid sidan av användarens namn
* Stäng av applikationen
* Av kommentera Admin Policy
* Tryck på publish under build fliken
* Nu är applikationen redo att användas.

## Api Endpoints

Http/get

**Api/Posts/{Id}**

Hämtar poster tråd id

* **ID**
* **DatePosted**
* **IsReported**
* **Posttext**
* **threadId**
* **userID**
* **Profilepicture**
* **UserBio**

Http/get Profanityfilter microservice

[**https://9chanprofanityfilter.azurewebsites.net/ProfanityFilter/{input**](https://9chanprofanityfilter.azurewebsites.net/ProfanityFilter/%7Binput)**}**

Tar in en Sträng och kollar efter engelska svordomar och skickar tillbaka en censurerad sträng

## Funktionalitet

#### Användarkonton

* Användarnamn
* Email
* Lösenord
* Presentation
* Profilbild
* Verifikation av konto via mail

#### Admin

* Skapa,ta bort, ändra kategori (OBS Poster och kommentarer tas bort)
* Skapa,ta bort, ändra underkategori (OBS Poster och kommentarer tas bort)
* Skapa grupper
* Ta bort gruppmedlemmar
* Skapa roller
* Göra användare till admin
* Ta bort admin rättigheter från användare
* Skapa Sticky Thread
* Avrapportera inlägg
* Ta bort rapporterade inlägg (OBS kommentarer för inlägget kommer också tas bort)

#### Meddelanden/Gruppmeddelanden

* Registrerade användare har möjlighet att skicka meddelanden till andra registrerade medlemmar, dessa meddelanden är krypterade och dekrypteras med AES algoritm Krypterings nycklarna finns i Azure secret vault om dessa behövs byta ut vid något tillfälle,
* Censurering av engelska svordomar

OBS gamla meddelanden måste använda sig av de krypteringsnycklar meddelandena var krypterade med

#### Forum

* Kategorielistan visar hur många subkategorier som finns i kategorien för översiktlighet
* Subkategorilistan visar hur många trådar som finns i subkategorien samt titel vem och när den senaste tråden skapades i subkategorin
* Trådlistan visar hur många poster som finns i tråden när den skapades samt vem och när den senaste posten gjordes i tråden.
* En tråd kan skapas av en registrerad användare med verifierat konto. En tråd måste ha en post för att skapas. Admin har möjlighet att göra tråden “sticky” så att den kommer högst upp i listan och är markerad för att lätt kunna se

* En post kan göras av en registrerad användare som har ett verifierat konto om posten innehåller engelska svordomar censureras de orden. En post kan innehålla en en bild genom att användaren lägger till en länk till en bild.

Posten kan gillas en gång av varje användare. Andra användare kan även rapportera posten till admin som kan ta ställning till om den ska tas bort eller ej,

En post kan kommenteras av andra användare och har samma funktionalitet som en vanlig post.

## Applicationens brister

#### Poster/Kommentarer

Dubbletter av bilder hämtas flera gånger så effektiviteten kan optimeras så att de inte hämtar samma bild flera gånger.

Verifikation av länkbilder behövs implementeras så att endast bilder kan länkas i posterna.

#### Profilbilder

Verifikation av filtyp vid uppladdning behövs implementeras

#### Routedata

Kryptering av data i adressfältet bör implementeras av säkerhetsskäl

## 

## 

## 

## Förbättringsförslag

* Text editor till Poster,kommentarer,användare presentation samt gruppmeddelanden och privata meddelanden.
* Thumbnail Preview av bilder
* signalR implementering för privata meddelanden
* Använding av razorcomponents och partialviews för återanvändning av tex tables
* Göra om Mypage så att den har responsivitet för mindre skärmar.
* Bryta ut funktionalitet till microservices med egna databaser
* En landingpage med översikt för användaren när den loggar in.
* Pagination för kategorier,subkategorier,trådar och poster
* En bättre design på sidan och hitta varför applicationen inte vill använda sig av css filen och flytta eventuell css till css filen/filer
* Se över dubblerad kod som finns

## Summering

Arbetet har gått väldigt bra och inte stött på några hinder. Dock så hade jag tacklat uppgiften annorlunda om jag hade fått göra om det. Under genomförandet så skrev jag kod för många olika sidor samtidigt istället för en sida i taget och tappade bort mig själv vad som var implementerat och vad som behövdes implementera och att inte implementera kod till saker som inte finns än även om det är en crud fortsättningsvis tills jag är mer bekväm med det. Fixa möjlighet för lokala secrets hade underlättat också när azures prov perioden avslutades