

Projektuppgift ASP.NET Core

Läranderesultat: Efter genomförd kurs ska den studerande ha kunskaper i/om:

3. Redogöra för hur en skyddad sida från obehörig åtkomst

Efter genomförd kurs ska den studerande ha färdigheter i att:

4. Tillämpa regler och rekommendationer för att sätta upp ett ASP.NET Core projekt som använder Razor Pages och/eller MVC på ett korrekt sätt
5. Utnyttja ASP.NET Razor Pages och/eller MVC för att bygga en webbapplikation enligt god programmeringssed.

Efter genomförd kurs ska den studerande ha kompetens för att:

6. Genomföra implementering av en webbapplikation med ASP.NET Core Razor Pages och/eller MVC med skydd mot obehöriga och enligt god programmeringssed

Övergripande

I denna inlämningsuppgift ska ni parprogrammera vidare på Event applikationen från tidigare. Vanliga **användare** ska fortfarande kunna se och gå med i olika roliga event som kommer att ske (efter att de har skapat ett konto/loggat in), men nu är sidan även till för två andra grupper: **administratörer** och **organisatörer**.

Läs igenom wireframes noggrant för att se hur sidan ska vara strukturerad

Authentication

För inloggning så kan ni skriva egna inloggningssidor eller använda/justera Identity paketets färdiga UI system. Nu ska Register och Log In knapparna fungera och man måste vara inloggad för att gå med i ett event (Ett enkelt sätt är att kräva inloggning på alla sidor förutom /Index sidan). Organisatörer och vanliga användare använder samma väg för att skapa ett konto, det är administratören som sedan gör om vanliga användare till administratörer.

Authorization

Det finns tre olika roller: **användare**, **organisatörer** och **administratörer**.

- Administratörer seedas in i databasen och kan inte skapas via hemsidan. De kan lägga till/ta bort organisatörsbefogenheten på alla användare.
- Organisatörer börjar med att skapa ett vanligt konto och sen ber en administratör att ge dem organisatörsbefogenheten. De kan se en lista på sina egna event och lägga till nya event (Valfritt tillägg: att lägga till hela CRUD).
- Vanliga användare saknar speciella befogenheter och kan bara gå med i event och se sina valda event (Valfritt tillägg: kunna avboka event, få påminnelser via e-mail mm.)

Ni bygger hela webbapplikationen som ett projekt i Visual studio eller VS Code, där samtliga delar som utgör applikationen kommer att finnas (både backend och frontend). Projektet ligger på ett gemensamt GitHub repo.

Deadline

Deadline inlämningsuppgift (fre 16/4)

G Krav

- En fungerade applikation med tillhörande databas skriven i Razor Pages och/eller MVC.
- Att alla wireframes finns implementerade
- Att man kan registrera sig och logga in som användare
- Att hemsidan anpassar sig beroende på ens behörigheter enligt detta dokumentet
- Lägga upp länk till projekt och .zip på PingPong (**alla i gruppen lägger upp samma**).

Enskild projektrapport (per person!)

G Krav

- En kort text som beskriver hur man använder Identity paketet för att identifiera och kontrollera besökares befogenheter.
- En kort rapport som sammanfattar vad du gjorde
- Lägga upp en pdf eller dylikt av rapporten på PingPong

VG Krav

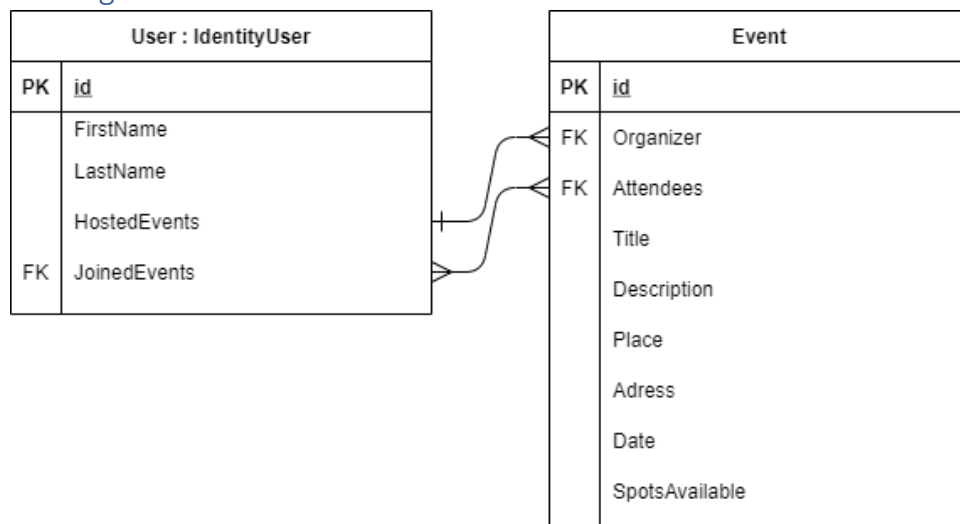
- En kort text i README där du enskilt skriftligen reflektera över, och analysera de lösningar som valts i grupprojektet vid uppsättning och implementering av ASP.NET Core Razor Pags och/eller MVC- projektet med skydd mot obehöriga

Beskrivning

Samma sidor som innan ska finnas, bara att "Register" och "Log In" knapparna nu ska länka till två riktiga sidor, samt att inloggade användare ska kunna logga ut.

Utöver detta så beskrivs här nedanför de nya sidor som ni ska skapa och de modifikationer ni måste göra.

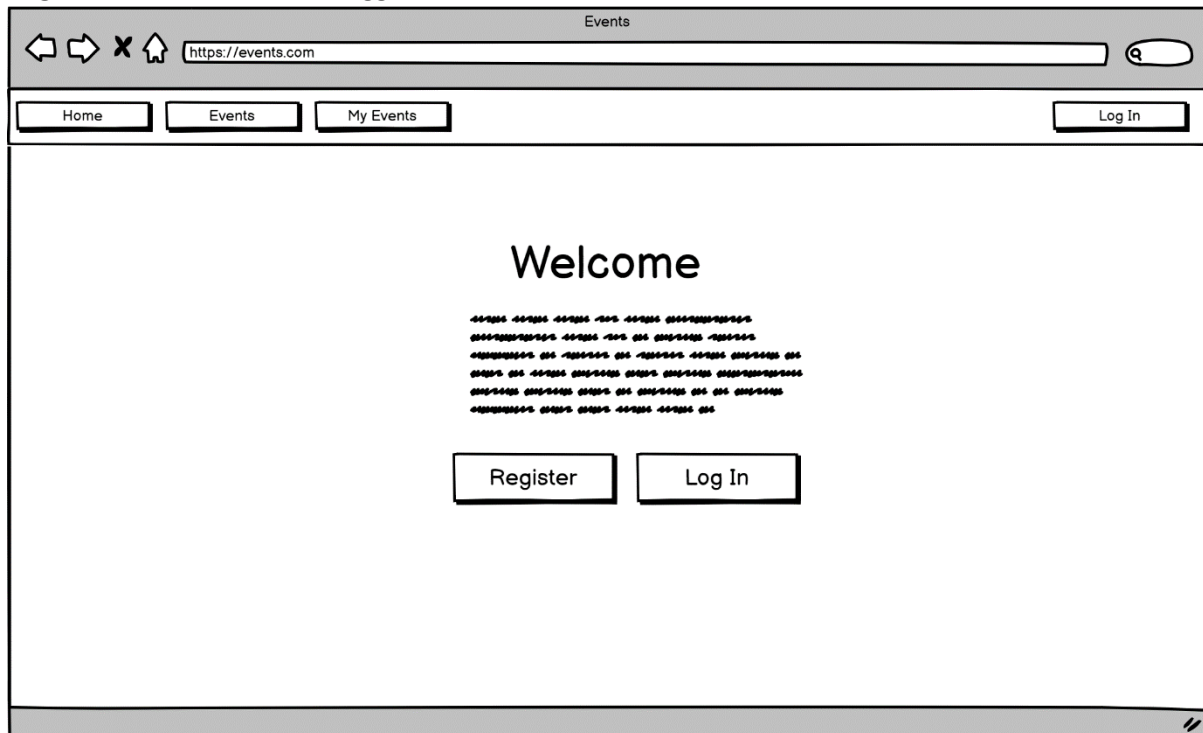
ER-diagramm



De tidigare modellerna Organizer och Attendee har kollapsat till en gemensam klass User som ärver av den underliggande IdentityUser. Särskiljningen mellan en organisatör och en vanlig användare sker nu via "Identity Roles" sytemet i stället. Använd [\[InverseProperty\] attributet](#) i Event tabellen för att särskilja mellan Organizer och Attendees referenserna till User tabellen.

Mockup

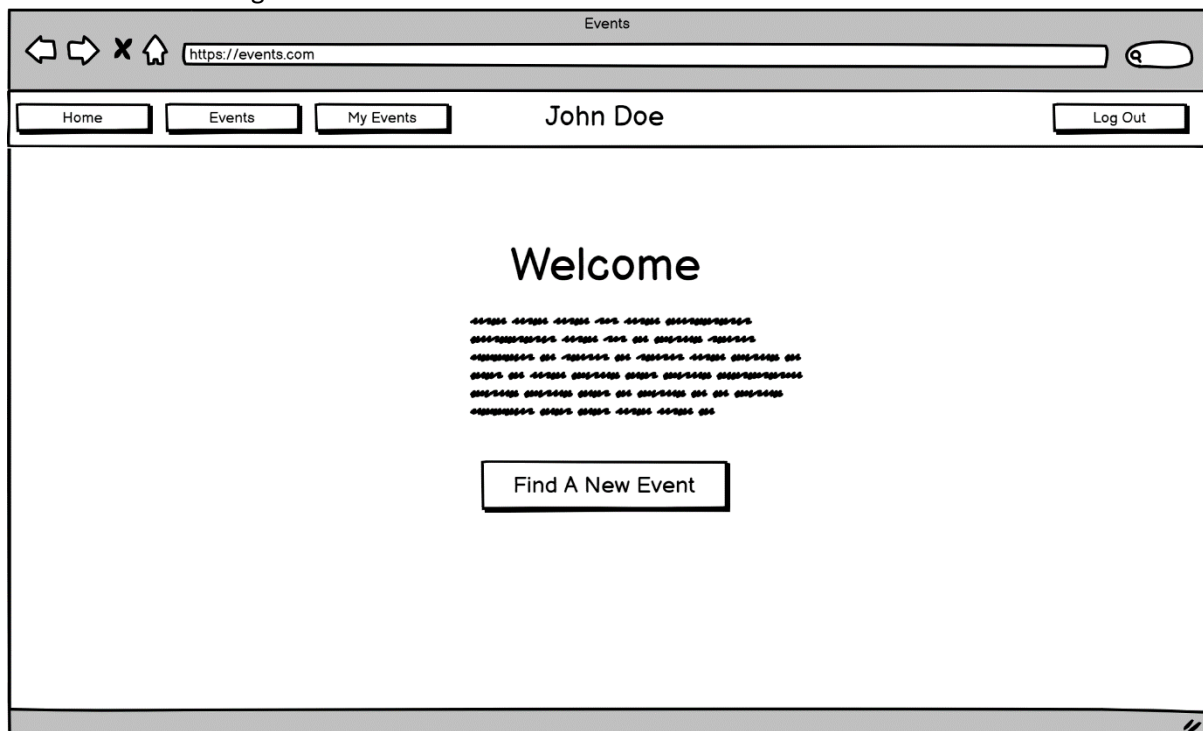
Alla utloggade besökare börjar vid samma skärm. Detta är den enda sidan förutom **Login**, **Logout** och **Register** sidorna som en oinloggad besökare kan använda.



När man väl har loggat in så visas användarens Namn/UserName alltid i Headern.

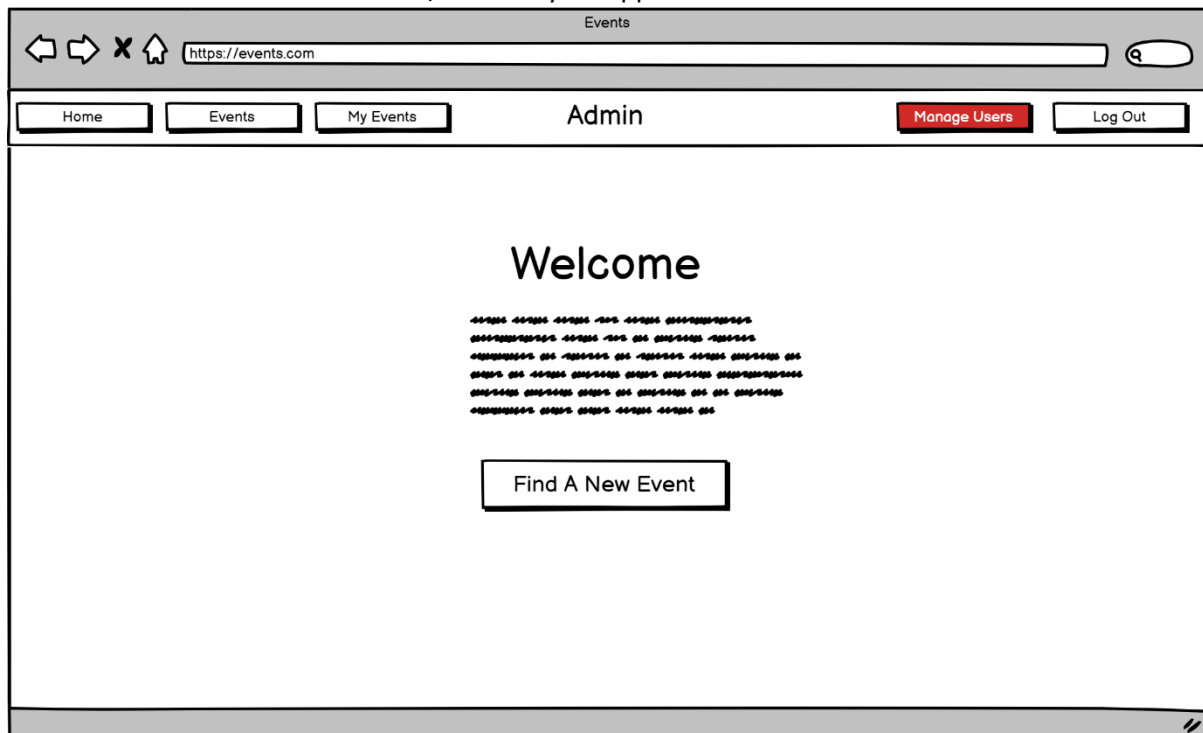
Vanlig användare

Vyn för en inloggad användare utan admin eller organisatörsbehörigheter. Alla sidor sen tidigare går att besöka som vanligt.

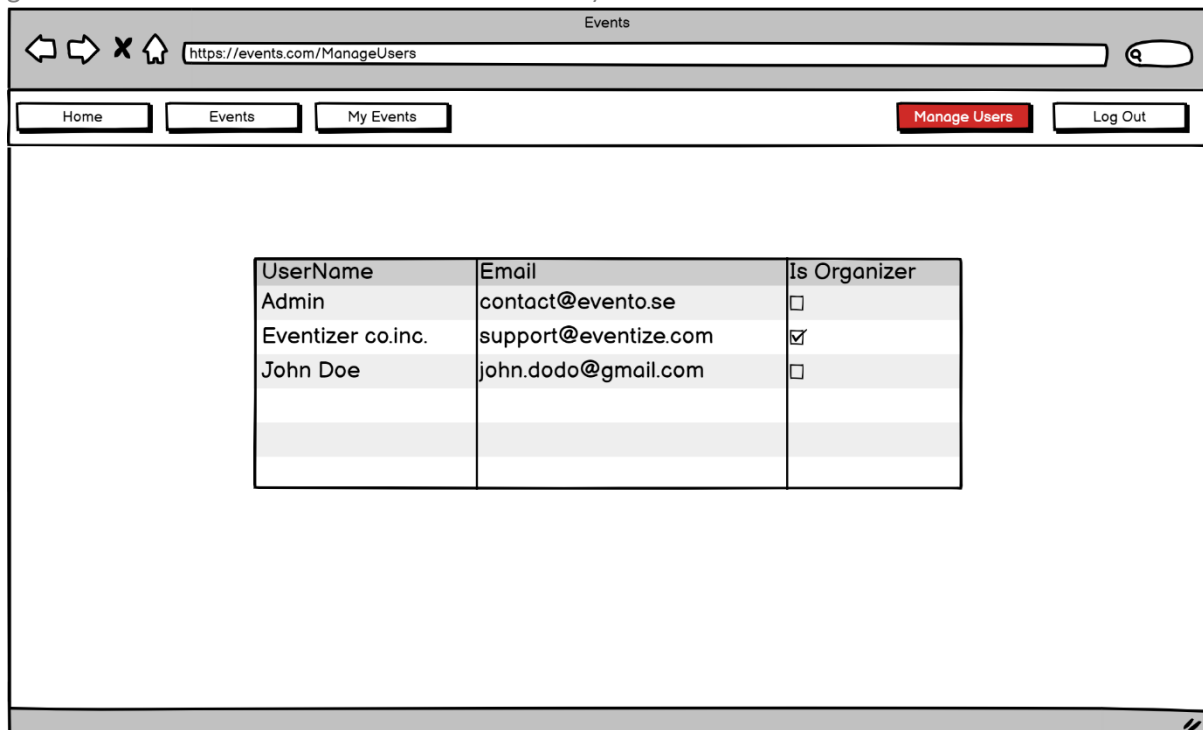


Admin

En länk bara administratörer kan se/besöka dyker upp i navbar.

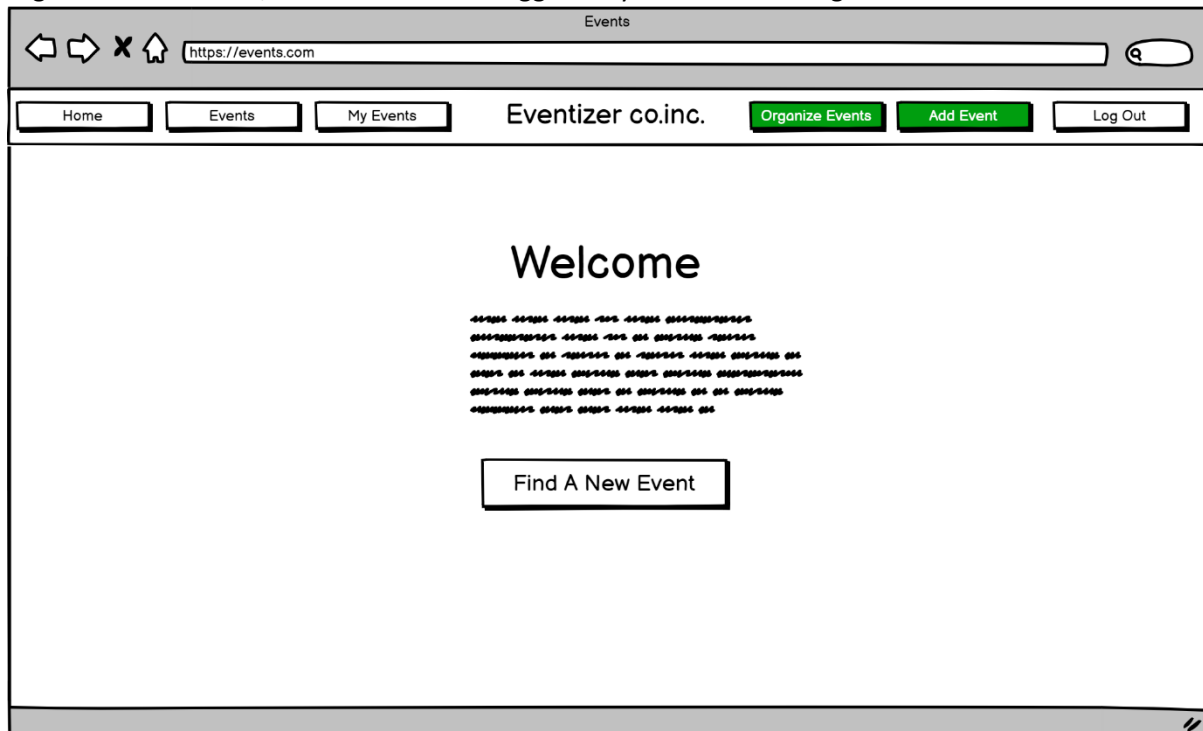


Här kan en inloggad administratör se alla nuvarande användare och checka i vilka som har organisatörsbehörigheten. (Valfritt tillägg: gör det möjligt att söka på UserName så att det lättare går att hitta den användare man vill administrera)

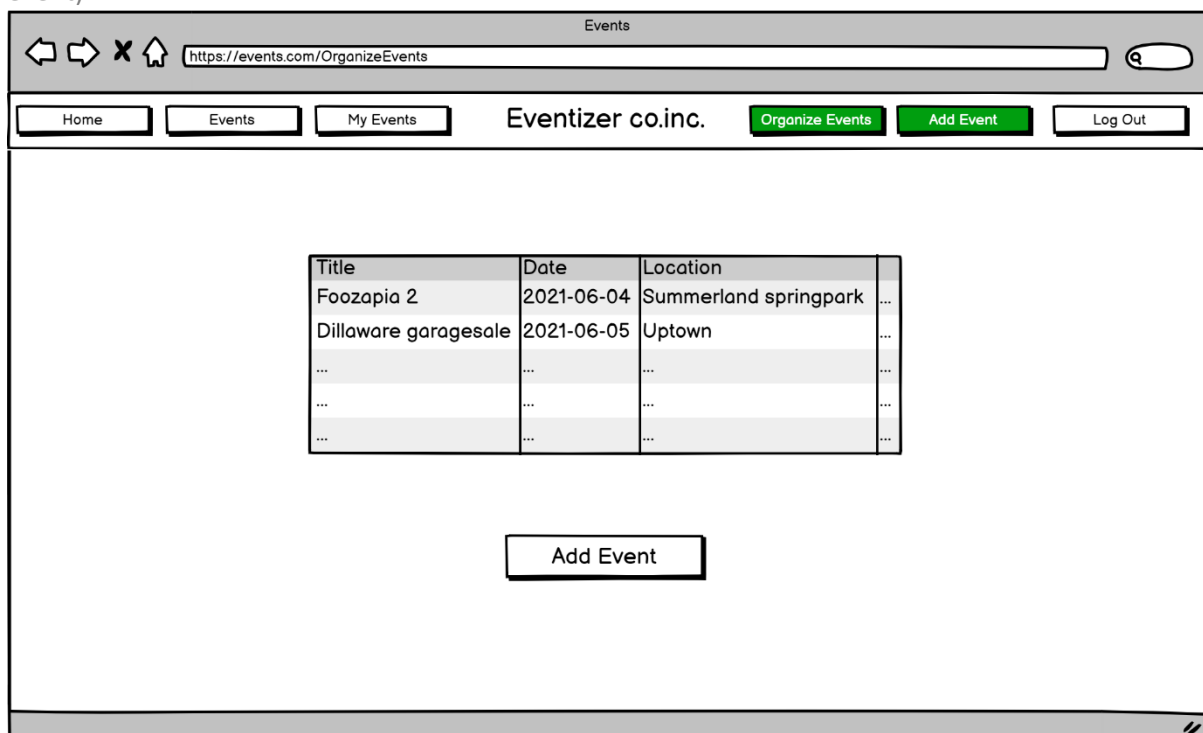


Organisatör

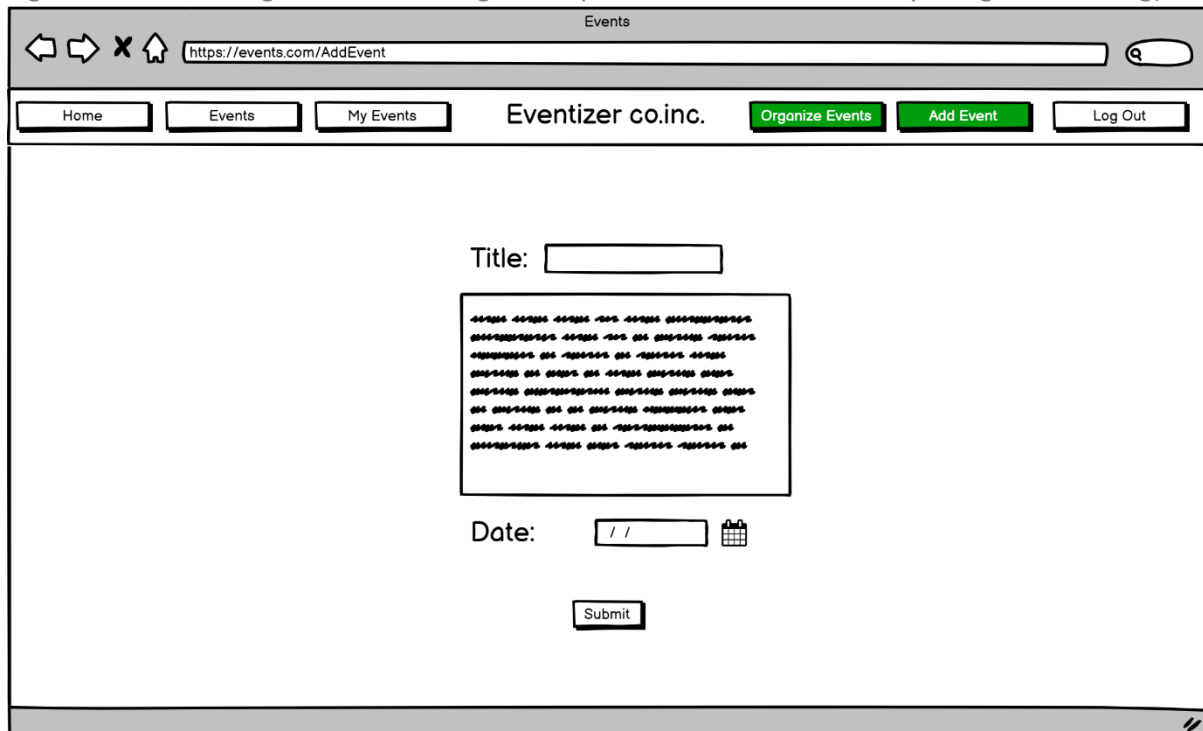
Det finns nu två länkar till två nya sidor. En för att överblicka alla event som den inloggade organisatören håller i, och en sida för att lägga till nya event under organisatörens namn.



Här finns det inte så mycket mer att göra än att se vilka events som är igång. (Valfria tillägg: lägg till så att events kan redigeras i efterhand, lägg till en sida som visar en lista på deltagare för ett specifikt event)



På denna sista sida så kan en organisatör lägga upp all info som behövs för att starta ett nytt event. (Valfritt tillägg: gör det möjligt att ladda upp en medföljande bild till ett event. Börja med att bara lagra en URL och sen gör att man kan lagra filen på servern om du vill ha en ytterligare utmaning)



Avgränsning

I den här uppgiften ligger fokus på att skapa en fungerande applikation i Razor Pages, fokus ligger inte på att skapa en snygg frontend med JavaScript, JQuery, etc. Ni får använda t.ex. Bootstrap men lägg inte fokus på det då det som bedöms i uppgiften är att ni har skapat en webbapplikation med den beskrivna funktionaliteten i ASP.NET.

Det finns tre alternativ till hur ni kan börja på projektet

1. Utgå från en av era inlämningsuppgifter
2. Utgå från facit koden (ni måste utgå från master branchen och bygga ut med Identity själva)
3. Skriva om koden med Identity templatens vald från start

Alla är lika ok, det viktiga är att jag ser att ni klarar av att använda och konfigurera Identity paketet

GitHub

Använd Git för att versionshantera koden. Committa varja dag och ofta. Ha er kod på ett **publikt GitHub repo** som ni länkar till när ni lämnar in eran .zip av projektet. **Både länk och .zip på PingPong.**

Tips

- Om det är svårt att hålla koll på alla sidor och hur de länkar till varandra, rita upp allt på ett diagram!
- Använd Identity scaffolding för att titta på hur Login, Logout, Register och RegisterConfirmed sidorna är uppbyggda. De är bra att studera för att lära sig hur man kan skriva bra if-satser i Razor eller hur man använder SignInManager och UserManager.

- Modifiera Seed metoden så att den tar in en IServiceProvider, då kan ni hämta SignInManager, RoleManager och UserManager tjänsterna precis som ni hämtar DbContext tjänsten. Sen kan ni använda samma metoder som används i Register sidan för att lägga till tex. en admin användare.
- RoleManager används för att etablera vilka roller som finns att välja mellan.