|  |
| --- |
| 2024-2025 |
| Vizsgaremek Dokumentáció |
| Pető Kristóf, Hegyes Csaba |

# **1 Fejlesztői Dokumentáció**

## **1.1 Specifikáció**

**1. Projekt neve**

* **Eventor**

**2. Célkitűzés**

* Olyan online platform létrehozása, ahol a felhasználók eseményeket hozhatnak létre, azokat kezelhetik, és más felhasználók regisztrálhatnak az eseményekre. A cél egy könnyen kezelhető, mobilbarát weboldal fejlesztése, ahol az eseményszervezés folyamata egyszerű és áttekinthető.

**3. Rendszer funkciói**

**3.1. Felhasználói funkciók**

* **Regisztráció és bejelentkezés**:
  + A felhasználók regisztrálhatnak és bejelentkezhetnek az oldalon.
  + Lehetőség harmadik fél általi bejelentkezésre (pl. Google, Facebook API integrációval).
* **Profilkezelés**:
  + A felhasználók módosíthatják profiljukat (pl. név, e-mail cím, profilkép).
  + Események szervezői további adatokkal (pl. szervezet neve) bővíthetik profiljukat.
* **Események létrehozása és szerkesztése**:
  + Az események alapadatainak megadása: cím, leírás, időpont, helyszín, kategória, maximális résztvevőszám.
  + Események képfeltöltési lehetősége.
  + Események szerkesztése és törlése.
* **Események keresése** :
  + Keresési lehetőség események között kategória, helyszín, dátum és kulcsszó alapján.
  + Szűrő opciók (pl. dátum szerinti rendezés, helyszín szerinti szűrés).
* **Eseményekre való regisztráció**:
  + A felhasználók egy gombnyomással regisztrálhatnak az eseményekre.
  + A regisztráció után e-mail értesítés küldése.
* **Meghívók küldése**:
  + Eseményszervezők e-mail meghívókat küldhetnek meghívottaknak.
* **Profiloldal és események nyomon követése**:
  + A felhasználók nyomon követhetik a saját eseményeiket (melyikre regisztráltak, melyikeket szervezik).

**3.2. Adminisztrátori funkciók**

* **Felhasználók kezelése**:
  + Adminok szerkeszthetik és törölhetik a felhasználói fiókokat.
* **Események moderálása**:
  + Az adminok figyelemmel kísérhetik és moderálhatják az eseményeket.
  + Tiltott események eltávolítása.

**4. Rendszer architektúrája**

**4.1. Frontend**

**Hegyes Csaba**

* **Technológiák**: HTML5, CSS3, JavaScript, Figma
* **Reszponzív design**: A weboldal mobil és tablet eszközökre optimalizált megjelenítése.

**4.2. Backend**

**Pető Kristóf**

* **Technológia**:
  + **Node.js (Express)** a szerver oldali logikához és API endpointok kezeléséhez.
* **Autentikáció**:
  + JWT (JSON Web Token) vagy session-alapú autentikáció a biztonságos bejelentkezés érdekében.
* **Adatkezelés**:
  + CRUD műveletek a felhasználók, események és regisztrációk kezelésére.

**4.3. Adatbázis**

* **Technológia**:
  + **MySQL**
* **Adattáblák**:
  + adattáblák.graphml

**4.4. Külső API integrációk**

* **Google Maps API**: Az események helyszínének megjelenítése térképen.
* **E-mail szolgáltatás (SendGrid free plan)**: E-mail értesítések és meghívók küldése.

**5. Biztonsági szempontok**

* **SSL titkosítás**: Az oldalon keresztül küldött adatok titkosítása.
* **Jelszó titkosítása**: A jelszavak biztonságos hash-elése.
* **XSS és CSRF védelem**: A felhasználói adatok megfelelő tisztítása és a kérések védelme cross-site scripting (XSS) és cross-site request forgery (CSRF) támadások ellen.

## **1.2 Adatbázis dokumentáció**

### **1.2.1 Adat reláció modell**

A computer screen shot of several rectangular objects

AI-generated content may be incorrect.

Ábra 1 yED terv képernyőkép  
  
**1.2.2 Adattáblák**

**User**id int primary key unique auto\_increment not null  
username varchar(30) unique null  
email varchar(100) unique not null  
password varchar(255) null (hash!!!)  
join\_date date default getdate()  
city\_id  
bio varchar(255)  
pfp varchar(255) (/assets/images/users/id)  
completed int (0,1)  
role (admin, user) varchar(12) permissions checked by static class

**Event**id primary key unique auto increment not null  
title varhcar(100)  
description varchar(1000)  
start\_date datetime not null  
end\_date datetime null  
visibility varchar(10) ("private" or "public")  
cover varhcar(255) (public/assets/images/events/id)  
city\_id  
location varchar(255)  
max\_response int null  
age\_limit bool  
date datetime not null default NOW()

**Cities**id primary key unique auto\_increment  
name varchar(100)  
county varchar(100)

**Category**id int primary key unique auto\_increment  
name varchar(30)  
description varchar(255)  
**EventUser**  
user\_id  
event\_id  
type ("view","respone", "invite")  
date datetime default NOW()

EventCategory  
category\_id  
event\_id

EventComment  
id primary key unique auto\_increment  
event\_id  
user\_id  
comment\_text varchar(255) not null  
super\_comment\_id int null (if null it is a top level comment)  
date datetime not null default NOW()

EventGalleryImage  
event\_id  
image varchar(255) (/public/assets/images/events/)

## **1.3 Funkció és REST API dokumentáció**

•Bázis URL: <https://eventor.petokrisa.hu/api/>  
•Protokoll: HTTPS   
•Formátum: JSON   
•Autentikáció: Bearer Token (JWT)

**HTML**

**GET "/"**

Returns text/html index.html

**GET "/event/:id"**

returns text/html event.html

**GET "/user/@:username"**

returns text/html profil.html

**GET "/user/@:username/edit"**

returns text/html profilEdit.html

**GET "/search"**

returns text/html search.html

**GET "/login"**

returns text/html login.html

**GET "/register"**

returns text/html register.html

**API**

**Error model**

(If an error occurs the http code will also show an error)

{

status: "error",

message: "Example error message!"

}

**Authorization header model**

Every endpoint that has (token) next to it requires the Authorization header in the request

Authorization: "Bearer <token>"

**1. User**

**GET "/api/user/@:username"**

response application/json:

{

id

username

joinDate

city {id, name, county}

bio?

pfp?

events: [

{

cover

title

startDate

endDate

}

]

respondedEvents: [

{

cover

title

startDate

endDate

}

]

}

**PUT "/api/user/@:username"**

city is the city's name, only data that is in the /api/cities is accepted body application/json:

bio

city

response application/json status, message:

**DELETE "/api/user/@:username" (token)**

If the user has permission it will delete the user body application/json:

{

password

}

response application/json status, message:

**GET "/api/user/@:username/pfp"**

returns the profilepicture as a file

**POST "/api/user/@:username/pfp" (token)**

updates user profilepicture body multipart/form-data:

pfp

**DELETE "/api/user/@:username/pfp" (token)**

deletes users profilepicture

**POST "/api/user/login"**

body application/json

{

username

password

}

response application/json

jwt

**POST "/api/user/register"**

only cities in /api/cities are allowed body application/json

{

username

password

email

city (name)

}

response application/json status, message

**GET "/api/user/verify"**

url query

username

id

pin

response redirect

**POST "/api/user/@:username/email" (token)**

body application/json

{

password

email

}

response application/json status, message

**GET "/api/user/@:username/email" (token)**

returns the email of the user (only admins and profile owners can see email beacuse it is not public information) response application/json

{

email

}

**POST "/api/user/@:username/password" (token)**

body application/json

{

password

newPassword

}

response application/json status, message

**2. Event**

**GET "/api/event/:id"**

response application/json

{

id

title

description?

startDate

endDate?

visibility

cover

city {id, name, county}

location

maxResponse

responseCount

gallery: [url, url, url]

ageLimit (boolean)

views

categories

}

**POST "/api/event" (token)**

body multipart/form-data :

title

description?

startDate

endDate?

visibility (public/private)

cover (file)

city (name)

location

maxResponse

gallery [] (files)

ageLimit (boolean)

categories []

response application/json status, message

**PATCH "/api/event/:id" (token)**

body multipart/form-data :

title?

description?

startDate?

endDate?

visibility? (public/private)

cover? (file)

city? (int id)

location?

maxResponse?

gallery []? (files)

ageLimit? (boolean)

response application/json status, message

**POST "/api/event/:id/respond" (token)**

Toggles user response to an event if the user has permission

response application/json status, message

**POST "/api/event/:id/invite" (token)**

Sends an invite to a user, and add them to the invited users of an event, if a user is invited already it will uninvite them body application/json:

{

invitedUserId

}

response application/json status, message

**3. Comment**

**GET "/api/event/:id/comment"**

body application/json:

[

{

id

user {id, username, pfpUrl}

text

replies [(same comment object)]

}

]

response application/json status, message

**POST "/api/event/:id/comment" (token)**

body application/json:

{

text

superCommentId?

}

response application/json status, message

**DELETE "/api/comment/:id" (token)**

deletes comment if user has permission response application/json status, message

**4. Search**

**GET "/api/search"**

query application/x-www-form-urlencoded:

q (text to search for)

categories? [] (list of categories to search for)

city?

startDate? (eg.: "2024-10-31 14:30", by default dates that are not in the past)

ageLimit? (boolean)

sortby? (name, startDate) (eg.: "name:desc" "startDate:asc")

page? (int, search only returns 12 per page)

response application/json:

[

{

id

title

cover

maxResponse

responseCount

startDate

city

ageLimit (boolean)

categories []

views

}

]

**GET "/api/search/popular"**

returns the most viewd events that are not expired response application/json: same as regular search

**GET "/api/search/recommended"**

returns events based on the users intrests response application/json: same as regular search

**5. Category and cities**

**GET "/api/category"**

q? (category name, search categories)

response application/json

[

{

id

name

description

}

]

**GET "/api/cities"**

query application/x-www-form-urlencoded:

q? (city name, search cities)

response application/json

[

{

id

name

county

}

]

## **1.4 Tesztek**

### **1.4.1 Manuális Tesztek**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Teszt Eset ID** | **Leírás** | **Lépések** | **Elvárt eredmény** | **Eredmény** |
| TE001 | Regisztráció város mező | 1 Nyitsd meg a /register oldalt  2 A város mezőbe kezdj el gépelni  3 Válassz ki egy várost a megjelent listából | 2 gépelt karakter után megjelenik egy lista a képernyőn | pass |
| TE002 | Regisztráció érvényes adatokkal | 1 Nyitsd meg a /register oldalt 2 Gépelj be egy érvényes felhasználó nevet, és email címet  3 add meg 2x ugyan azt a jelszót | AZ oldal átírányít a bejelentkezés oldalra,  Emailben verifikációs linket kapsz | pass |
| TE002 | Regisztráció érvénytelen jelszóval | 1 Nyitsd meg a /register oldalt 2 adj meg érvényes adatokat mindenhova  3 a 2. Jelszó mezőn írj be egy eltérő jelszót | Hibát ír az oldal “A jelszavak nem egyeznek” | pass |
| TE003 | Regisztráció érvénytelen várossal | 1 Ismételd meg a TE002 lépéseit, de a város mezőbe gépelj egy nemlétező város nevet | Hibát ír az oldal “Nem létezik ilyen nevű város” | pass |
| TE004 | Bejelentkezés érvénytelen adatokkal | 1 Nyitsd meg a /login oldalt 2 Írj be egy létező felhasználó nevet, de rossz jelszót | Hibát ír az oldal “Hibás jelszó” | pass |
| TE005 | Bejelentkezés érvénytelen felhasználónévvel | 1 Nyitsd meg a /login oldalt 2 írj be egy nem létező felhasználó nevet | Hibát ír az oldal “Ilyen nevű felhasználó nem létezik” | pass |
| TE006 | Bejelentkezés érvényes adatokkal | 1 Nyitsd meg a /login oldalt 2 Nyitsd meg az emailben kapott linket hogy hitelesítsd a fiókod  3 írd be az érvényes felhasznélói adataidat | Az oldal átirányít a profilodra | pass |
| TE007 | Profilkép változtatása | 1 Nyitsd meg a fiók beállításokat a jobboldali menüből  2 Válassz ki egy file-t  3 nyomd meg a feltöltés gombot | Az oldal kiírja hogy “profilkép frissítve” majd 2 másodperc után frissít | pass |
| TE008 | Adatok változtatása | 1 írd át a városod, és a bio-d 2 Nyomj a frissítés gombra | Az oldal frissíti az adataidat | pass |
| TE009 | Új esemény létrehozása | 1 a jobboldali menüből választd ki az új esemény menüpontot  2 töltsd ki a mezőket  3 válassz 1 borítóképet és legalább 1 galéria képet | Az oldal átirányít az újonnan létrehozott eseményre | pass |
| TE010 | Esemény törlése | 1 az újonnan létrehozott eseményen felül nyomj a törlés gombra 2 a felugró ablakon nyomj az ok gombra | Az oldal átirányít a fiókodra | pass |

### **1.4.2 Automata Tesztek**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Teszt Eset ID** | **Leírás** | **Lépések** | **Elvárt eredmény** | **Eredmény** |
| AT001 | Regisztráció Rövid jelszóval | 1 Nyitsd meg a /register oldalt  2 Gépeld be az adataidat, de a jelszó legyen kevesebb mint 8 karakter | 2 gépelt karakter után megjelenik egy lista a képernyőn | pass |
| AT002 | Regisztráció érvényes adatokkal | 1 Nyitsd meg a /register oldalt 2 Gépelj be egy érvényes felhasználó nevet, és email címet  3 add meg 2x ugyan azt a jelszót | AZ oldal átírányít a bejelentkezés oldalra,  Emailben verifikációs linket kapsz | pass |
| AT002 | Regisztráció érvénytelen jelszóval | 1 Nyitsd meg a /register oldalt 2 adj meg érvényes adatokat mindenhova  3 a 2. Jelszó mezőn írj be egy eltérő jelszót | Hibát ír az oldal “A jelszavak nem egyeznek” | pass |
| AT003 | Regisztráció érvénytelen várossal | 1 Ismételd meg a TE002 lépéseit, de a város mezőbe gépelj egy nemlétező város nevet | Hibát ír az oldal “Nem létezik ilyen nevű város” | pass |
| AT004 | Bejelentkezés érvénytelen adatokkal | 1 Nyitsd meg a /login oldalt 2 Írj be egy létező felhasználó nevet, de rossz jelszót | Hibát ír az oldal “Hibás jelszó” | pass |
| AT005 | Bejelentkezés érvénytelen felhasználónévvel | 1 Nyitsd meg a /login oldalt 2 írj be egy nem létező felhasználó nevet | Hibát ír az oldal “Ilyen nevű felhasználó nem létezik” | pass |
| AT006 | Bejelentkezés érvényes adatokkal | 1 Nyitsd meg a /login oldalt 2 Nyitsd meg az emailben kapott linket hogy hitelesítsd a fiókod  3 írd be az érvényes felhasznélói adataidat | Az oldal átirányít a profilodra | pass |
| AT007 | Profilkép törlése | 1 Nyitsd meg a fiók beállításokat a jobboldali menüből  2 nyomd meg a törlés gombot | Az oldal kiírja hogy “Sikeres törlés” majd 2 másodperc után frissít | pass |
| AT008 | Adatok változtatása | 1 írd át a városod, és a bio-d 2 Nyomj a frissítés gombra | Az oldal frissíti az adataidat | pass |
| AT009 | Új esemény létrehozása | 1 a jobboldali menüből választd ki az új esemény menüpontot  2 töltsd ki a mezőket  3 válassz 1 borítóképet és legalább 1 galéria képet | Az oldal átirányít az újonnan létrehozott eseményre | fail |
| AT010 | Esemény törlése | 1 az újonnan létrehozott eseményen felül nyomj a törlés gombra 2 a felugró ablakon nyomj az ok gombra | Az oldal átirányít a fiókodra | fail |

A screenshot of a computer

AI-generated content may be incorrect.

Ábra 2 Automata teszt eredmények  
  
Az automata tesztekből egy kivételével, mindegyik lefutott, az esemény létrehozásánál viszont mindig visszairányít az oldal a bejelentkezésre. A profil oldalnál sikerült a BeforeEach() függvényben bejelentkezni mielött a teszteket futtatom, de az esemény létrehozása elött a bejelentkezés ugyanazokkal az adatokkal, ugyan úgy nem fut le.

# **2 Felhasználói dokumentáció**

## **2.1 Főbb funkciók bemutatása**

## **2.1.1 Regisztráció és Bejelentkezés**

Az oldalon a regisztrációhoz felhasznált adatok a következők: Felhasználónév, Email, Jelszó, Város.   
A felhasználónév és az email egyedinek kell lennie minden felhasználónak, ezenkívül a felhasználó nevet nem lehet megváltoztatni, ugyanis az alapján lehet beazonosítani az adott felhasználót.

A screenshot of a phone

AI-generated content may be incorrect.

Ábra 3 Regisztrációs kérdőív

A hagyoményos regisztráción kívül lehetőség van Google fiókkal való belépéshez, így a felhasználónevet, az emailcímet és a profilképet a google kezeli.

A vársokhoz csak létező, Magyar város neveket lehet beírni, különben a rendszer nem fogja elfogadni. Ehez segítség képpen minden város beviteli mezőnél 2 karakter beírása után megjelenik egy lista, a legjobban hasonlító város nevekkel. A screenshot of a computer

AI-generated content may be incorrect.

Ábra 4 Város beviteli mező

Erre azért van szükség, mert az adatbázisba minden városhoz el van tárolva a megye is amiben van, ezért a keresésnél lehetséges megyékre szűrni, nem cask városokra

A jelszóra 3 kikötés van:

* 1. Hosszabbnak kell lennie mint 7 karakter (legalább 8)
  2. Tartalmaznia kell számokat
  3. A 2 jelsóbeviteli mezőnek egyeznie kell

A close up of a sign

AI-generated content may be incorrect.



A close up of a sign

AI-generated content may be incorrect.

A regisztráció után, emailben kap a felhasználó egy linket, amit meg kell nyitnia mielött be tud jelentkezni az oldalra

Példa:  
*https://eventor.petokrisa.hu/api/user/verify?username=asdasd12&id=29&pin=1871052*

Ez abban segít hogy ne lehessen nem létező emailcímmel felregisztrálni az oldalra.

### **2.1.2 Bejelentkezés**

A screenshot of a computer screen

AI-generated content may be incorrect.

Ábra 5 Bejelentkezés kérdőív

A bejelentkezéshez cask a felhasználónévre és az emailícmre van szükség, de itt is lehetőség van a google bejelentkezésre.

### **2.1.3 Fiók**

A screenshot of a computer

AI-generated content may be incorrect.

Ábra 6 Fiók beállítások

Az első menüpontban a profilképet lehet megváltoztatni, vagy kitörölni.  
Alatta a városodat, és a leírásodat, amit szerkesztés után a frissítés gombbal frissíthetünk. Az emailcímet és a jelszót, cask a jelelnlegi jelszó beírásával lehet megváltoztatni biztonsági okokból.   
Emellett a fiókot teljesen kilehet törölni, de ezt is cask a jelszó megadásával

### **2.1.4 Esemény**

Az oldal fő funkciója, lehetővé teszi hogy létrehozzunk eseményeket, képeket töltsünk fel hozzájuk, meghívjunk rá embereket, és regisztráljunk.

A close up of a pair of glasses

AI-generated content may be incorrect.

Ábra 7 Esemény oldal 1. fele

A person taking a selfie

AI-generated content may be incorrect.

Ábra 8 Galéria

A white rectangular object with a black border

AI-generated content may be incorrect.

Ábra 9 Kommentekw

### **2.1.5 Profil**

A profil oldalon látható a saját vagy más felhasználók profilja, érdeklődési körei, leírása, eseményeik, és regisztrációik.

A screenshot of a social media account

AI-generated content may be incorrect.

Az érdeklődési körök automatikusan kerülnek fel a profilra, a regisztrált események kategóráit összesíti az oldal. Érdeklődési kört törölni cask úgy lehet, ha leiratkozunk arról az eseményről amik azokkal a kategóriákkal vannak ellátva.

### **2.1.6 Események létrehozása**

A screenshot of a computer

AI-generated content may be incorrect.

Ábra 10 Új eseméyn létrehozása

Az eseményszervező oldalon található eseménylétrehozási űrlap lehetőséget biztosít a felhasználók számára új események rögzítésére. Az alábbi mezők állnak rendelkezésre az esemény megfelelő adatainak megadásához:

**Kötelező Mezők:**

1. **Esemény Neve** – Az esemény megnevezése, amely megjelenik a listázásban.
2. **Kezdő Dátum** – Az esemény kezdési időpontja (ÉÉÉÉ. HH. NN. –:– formátumban).
3. **Város** – A város megadása, ahol az esemény megrendezésre kerül.

**Opcionális Mezők:**

1. **Leírás** – Az esemény rövid ismertetése.
2. **Kategóriák** – Az esemény témájának megfelelő kategóriák kiválasztása (pl. koncert, sport, fesztivál, online stb.).
3. **Vég Dátum** – Az esemény befejezési időpontja.
4. **Helyszín** – Az esemény pontos helyszíne.
5. **Férőhely** – A résztvevők maximális száma.
6. **Korhatár** – Beállítható korhatár megkötés az eseményhez.

**Fájlok Feltöltése:**

* **Borítókép Feltöltése** – Egy kép kiválasztása, amely az eseményhez kapcsolódóan borítóképként jelenik meg.
* **Galéria Képek Feltöltése** – További képek feltöltése az esemény galériájához.

### **2.1.7 Esemény szerkesztése**

A screenshot of a computer

AI-generated content may be incorrect.

Ábra 11 Szerkesztés űrlap

Az eseményszervező oldalon az eseményszerkesztési űrlap lehetőséget biztosít a már létrehozott események adatainak módosítására. Az alábbi mezők és funkciók érhetők el az esemény szerkesztése során:

**Mezők és Funkciók:**

1. **Borítókép Feltöltése és Törlése** – Lehetőség van egy új borítókép feltöltésére, valamint a meglévő borítókép törlésére.
2. **Cím**\* – Az esemény neve módosítható.
3. **Leírás** – Az esemény ismertető szövegének frissítése.
4. **Kategóriák** – Az eseményhez tartozó kategóriák szerkesztése, új kategória hozzáadása.
5. **Kezdés és Végzés Dátuma**\* – Az esemény időpontjának módosítása (ÉÉÉÉ. HH. NN. –:– formátumban).
6. **Város**\* – Az esemény városának frissítése.
7. **Helyszín** – Az esemény pontos helyszínének módosítása.
8. **Maximális Létszám** – A résztvevők maximális számának beállítása.

**Galéria Képek Kezelése:**

* Az eseményhez tartozó galéria képek szerkeszthetők, új képek tölthetők fel, illetve meglévő képek törölhetők.

Kategóriák hozzáadásához egy felugró ablak használható  
A screenshot of a computer

AI-generated content may be incorrect.

Ábra 12 Ketegória szerkesztés

A screenshot of a computer

AI-generated content may be incorrect.

Ábra 13 Kép feltöltés

A screenshot of a phone

AI-generated content may be incorrect.

Ábra 14 Kép törlése

Tartalomjegyzék

[**1 Fejlesztői Dokumentáció** 1](#_Toc193965156)

[**1.1 Specifikáció** 1](#_Toc193965157)

[**1.2 Adatbázis dokumentáció** 5](#_Toc193965158)

[**1.2.1 Adat reláció modell** 5](#_Toc193965159)

[**1.3 Funkció és REST API dokumentáció** 7](#_Toc193965160)

[**1.4 Tesztek** 19](#_Toc193965161)

[**1.4.1 Manuális Tesztek** 19](#_Toc193965162)

[**1.4.2 Automata Tesztek** 21](#_Toc193965163)

[**2 Felhasználói dokumentáció** 24](#_Toc193965164)

[**2.1 Főbb funkciók bemutatása** 24](#_Toc193965165)

[**2.1.1 Regisztráció és Bejelentkezés** 24](#_Toc193965166)

[**2.1.2 Bejelentkezés** 26](#_Toc193965167)

[**2.1.3 Fiók** 27](#_Toc193965168)

[**2.1.4 Esemény** 28](#_Toc193965169)

[**2.1.5 Profil** 29](#_Toc193965170)

[**2.1.6 Események létrehozása** 30](#_Toc193965171)

[**2.1.7 Esemény szerkesztése** 31](#_Toc193965172)