**Eötvös Loránd Tudományegyetem**

Informatikai Kar

Média- és Oktatásinformatikai Tanszék

**Modern Internetes Képes Kommunikáció**

**Témavezető Szerző**

Dr. Horváth Győző Franta Áron

Egyetemi docens Programtervező informatikus

**Budapest, 2022.**

„*Nem az a feladat, hogy minden hallgatónknak mást, hanem az, hogy mindegyiknek eleget adjunk.*”

- EötvösLoránd

Eredetiségi Nyilatkozat

Tartalom

[1. Bevezetés 5](#_Toc122704015)

[1.1 Az oldal leírása, látványterv 5](#_Toc122704016)

[1.2 A használt technológiák 8](#_Toc122704017)

[2. Felhasználói dokumentáció 9](#_Toc122704018)

[2.1 A program használata saját eszközön 9](#_Toc122704019)

[2.1.1. Telepítés 9](#_Toc122704020)

# 1. Bevezetés

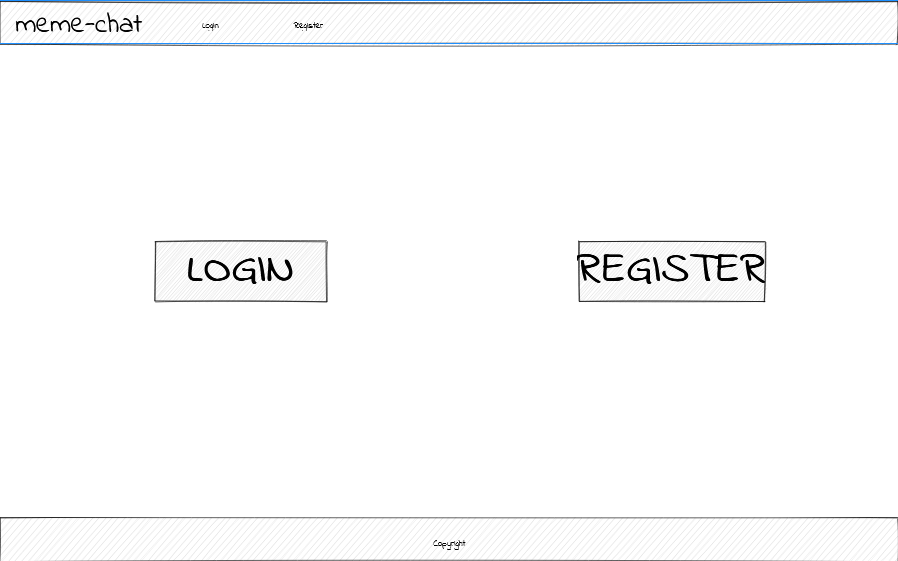
## 1.1 Az oldal leírása, látványterv

Szakdolgozatom egy modern eszközökkel készített internetes chat alkalmazást valósít meg, melyben az online kultúra aktív részévé vált mémekkel lehet kommunikálni. A mémek kis méretű képek melyekre szöveget helyeznek, ezek a szövegek általában egyszerű rövid mondatok, amely az adott szubkultúrában járatos embereknek tartalmat, hordoznak általában szórakoztató módon.

Mivel én is rendszeresen használom ezt a formátumot kommunikációra ezért érdekes kihívásnak tartottam egy olyan oldal megalkotását, ahol képeket manipulálva, különböző sablonokat kitöltve lehet kommunikálni.

Az oldalt egy egyoldalas (one-page) applikációként képzeltem el, amely egy főoldalból, egy bejelentkező és egy regisztrációs felületből áll, valamint egy sablonfeltöltő oldalból, egy chat listából és egy megjelenítő és küldőfelületből áll.

A főoldal nem bejelentkezett felhasználó esetén egyszerűen két gombot mutat, egy bejelentkezés és egy regisztráció.



1.1.1. ábra – A főoldal látványterve bejelentkezés nélkül

Bejelentkezett felhasználó esetén a főoldal a chateket listázó felületre mutat, amelyen megjelennek a chatek és van lehetőség új chatet létrehozni.

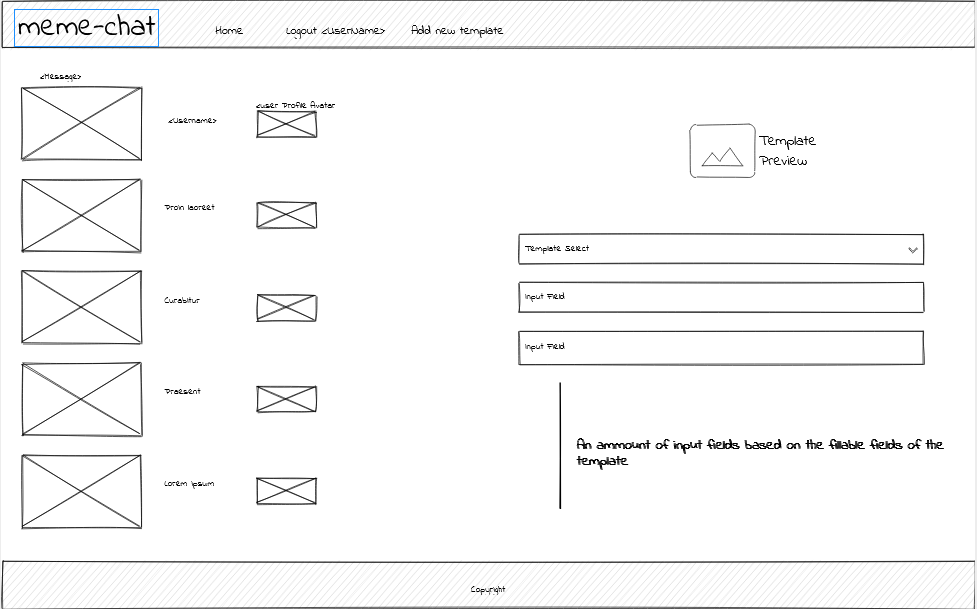
A képen szöveg látható

Automatikusan generált leírás

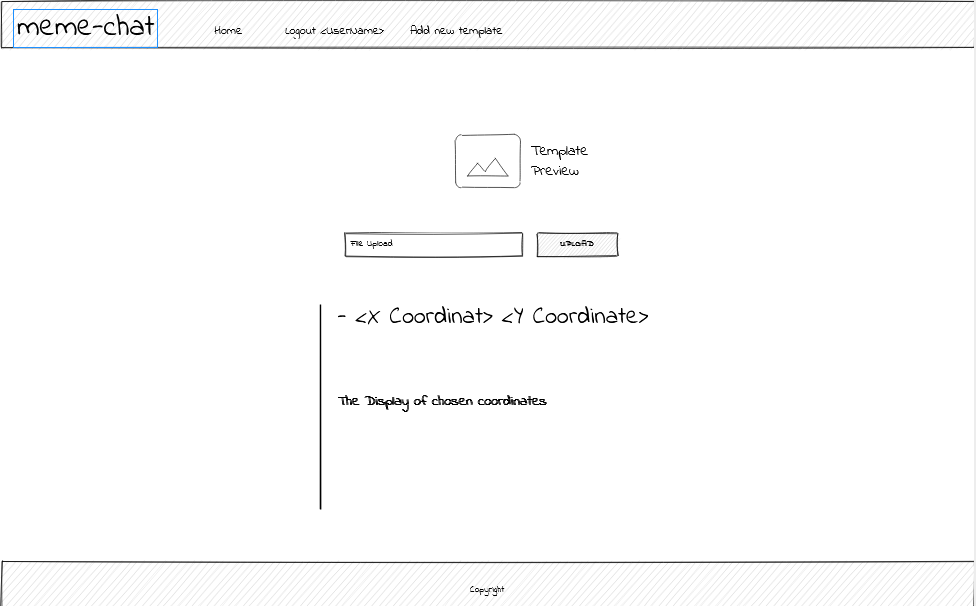
1.1.2. ábra – A chat lista látványterve

A chat oldalon láthatóak a chatbe küldött üzenetek, a küldő felhasználó és a felhasználó email címéhez tartozó profilkép. Mellette pedig látható a küldő felület. Ezen a felületen a felhasználó egy lenyíló menüből választhat az elérhető sablonok közül, a sablon kitölthető mennyiségű mezői szerint pedig megjelennek kitölthető szövegmezők. Ezek a 1.1.3. ábrán láthatók.

A sablon feltöltő felület (1.1.4. ábra) egy szövegmezőből áll, ahol elnevezhetjük a sablon, továbbá egy feltöltő mezőből. Ha feltöltünk egy képet a kép előnézete megjelenik. A képre kattintva a kép alatt megjelennek a kattintási koordináták. Ezek lesznek azok a mezők amiket a küldő oldalon kitöltünk.



1.1.3. ábra – A chat megjelenítő oldal látványterve



1.1.3. ábra – A sablon feltöltő oldal látványterve

## 1.2 A használt technológiák

Az oldal megvalósításához teljes mértékben modern webes technológiákat használtam a moduláris működés és könnyű fejleszthetőség érdekében. Fontosnak tartom az ingyenes és open-source technológiák használatát ezért a fullstack alkalmazásomhoz tartozó keretrendszereket és az adatbázist is ez alapján választottam.

Adatbázisnak egy nem SQLt használó adatbázist választottam, a MongoDB-t, egyszerű használata és akár ingyenesen is használható online felhő alapú hosting rendszere miatt. A Mongoose keretrendszer használatával, a backenden könnyen lehet csatlakoztatni, továbbá nem igényli az összetett adatkapcsolatokat.

A backend server REST API-okon keresztül kommunikál a frontend-el, ehhez az egyetemi karrierem során megismert Fastify helyett az Express.js keretrendszert választottam, bár a Fastify gyorsabb és emelett szebb syntaxal rendelkezik, az Express.js ügyesebben kezeli az esetleges üres adatokat és mivel nagyobb a felhasználó bázisa az interneten ezért rengeteg dokumentáció és segédanyag elérhető, ami nagyban megkönnyítette, hogy az olyan funkciókat is probléma nélkül használjam amik ritkábbak és nem feltétlenül egyértelműek.

A frontend keretrendszernek a META által kiadott React keretrendszert választottam. A célom ezzel a választással, a könnyű komponensekből álló felépítés és a jól dokumentált működés volt, ezek mellet közre játszott továbbá az hogy a React pár éve bevezette a funkcionális komponenseket melyek lecserélik az eddigi osztály alapú komponenseket. Ezek könnyebben tesztelhetőek mivel nincs saját állapotuk és teljes létezés alatt létező hook. Ehelyett ezek a JSX-et (JavaScript XML) visszatérítő java script függvények egyszeri propertyket használnak, emiatt használatba vettem a Reduxot hogy központosítsam az adattárolást egyetlen egy központi applikáció állapotba.

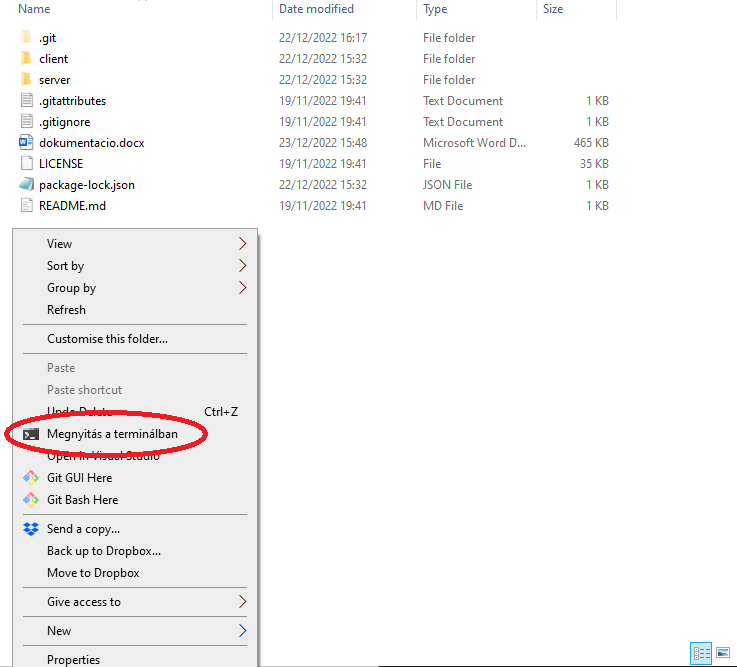
# 2. Felhasználói dokumentáció

## 2.1 A program használata saját eszközön

### 2.1.1. Telepítés

Első sorban a `node --version` segítségével ellenőrizzük hogy elérhető e a legfrissebb Node verzió eszközünkön, hiszen a Node Package Managert fogjuk használni (továbbiakban: npm). Ha ezzel megvagyunk akkor letölthetjük a szakdolgozat mappáját, ez tartalmaz egy server és egy client mappát, ezeknek mindegyike tartalmaz egy szükséges package-lock.json fájlt ami alapján az npm tudni fogja mely csomagokat kell telepíteni.

Nyissunk egy terminált a gyökér mappában:



2.1.1.1. ábra – Terminál nyitása Windows operációs rendszerben

Ezután a