„Peták.hu”  
Fejlesztői dokumentáció

[„Peták.hu” Fejlesztői dokumentáció 1](#_Toc132632787)

[Alapkövetelmények 2](#_Toc132632788)

[Tudnivalók 2](#_Toc132632789)

[Futtatáshoz szükséges szoftverek: 2](#_Toc132632790)

[Futtatáshoz kötelező node modulok: 2](#_Toc132632791)

[Futtatás 2](#_Toc132632792)

[Mappastruktúra 2](#_Toc132632793)

[Projektmappa 2](#_Toc132632794)

[A mappák: 2](#_Toc132632795)

[A fájlok: 2](#_Toc132632796)

[Backend 2](#_Toc132632797)

[A mappák: 3](#_Toc132632798)

[A fájlok, és tartalmuk: 3](#_Toc132632799)

[Frontend 3](#_Toc132632800)

[A mappák 4](#_Toc132632801)

[A fájlok, és tartalmuk 4](#_Toc132632802)

[Az alkalmazás 5](#_Toc132632803)

[Kommunikáció 5](#_Toc132632804)

[Adatbázis 5](#_Toc132632805)

[Backend 5](#_Toc132632806)

[Frontend 5](#_Toc132632807)

[Authorizáció 5](#_Toc132632808)

[Tesztek 5](#_Toc132632809)

[Továbbfejlesztés 5](#_Toc132632810)

[Frontend 5](#_Toc132632811)

[Backend 5](#_Toc132632812)

# Alapkövetelmények

## Tudnivalók

A fejlesztés Visual Studio Code (v 1.76.0) IDE-vel készült.

A webalkalmazás backendje Node.js-el (18.13.0), frontendje Svelte (Svelte3) keretrendszer segítségével jött létre. A Frontend fejlesztésénél TypeScript lett használva.

A frontend a backend-el a „socket.io”, és „axios” npm modulokkal kommunikál.

## Futtatáshoz szükséges szoftverek:

* node (16.13.0, vagy újabb)
* npm (8.19.3, vagy újabb)
* Visual Studio Code (1.67.2, vagy újabb)
* XAMPP (3.2.4, vagy újabb)
* Firefox (111.0.1, vagy újabb), vagy
* Google Chrome (111.0.5.5, vagy újabb)

## Futtatáshoz kötelező node modulok:

* socket.io (4.6.0, vagy újabb)
* socket.io-client (4.6.0, vagy újabb)
* tinro (0.6.12, vagy újabb)
* bootstrap (5.2.0, vagy újabb)
* bootstrap-icons (1.10.3)
* crypto-js (4.1.1, vagy újabb)
* chart.js (4.2.1, vagy újabb)
* axios (1.2.4, vagy újabb)
* sass (1.5.4, vagy újabb)
* express (4.18.2, vagy újabb)
* jsonwebtoken (9.0.0)
* multer (1.4.5, vagy újabb)
* mysql (2.18.1, vagy újabb)
* cors (2.8.5, vagy újabb)
* nodemailer (6.9.1, vagy újabb)
* path (0.12.7, vagy újabb)
* dotenv (16.0.3, vagy újabb)
* és ezek dependenciei

## Futtatás

A program futtatásához a backend és frontend mappáiban telepítenünk kell a node modulokat. Ezt az „npm i” paranccsal tehetjük meg.

A szerver indítása előtt érdemes beimportálni a DB mappában található sql fájlt a phpmyadmin felületen.

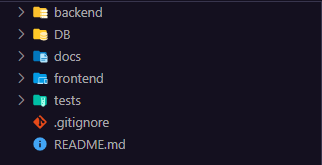
A szervert és a frontendet is az „npm start” paranccsal, a fejlesztői környezetet az „npm run dev”-el indítjuk el.

A backend a <http://localhost:8080>, a frontend <http://localhost:5173> címeken érhetőek el, miután futtatjuk őket.

# Mappastruktúra

## Projektmappa

A projekt mappájának megnyitásakor a következő mappákat láthatjuk.



. ábra Fő mappastruktúra

### A mappák:

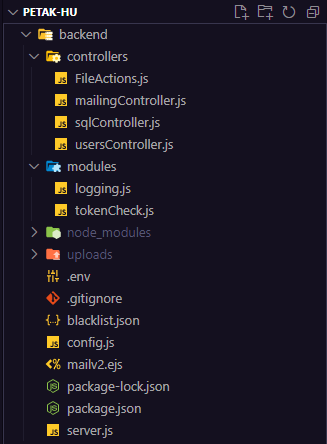
* backend
  + Ez a mappa tárolja a backend összes fájljait.
* DB
  + Ez a mappa tárolja az alkalmazásban használt adatbázist.
* docs
  + Ebben a mappában találhatóak a projekt dokumentumai  
    (pl.: programterv, fejlesztői dokumentáció, felhasználói dokumentáció, stb.)
* frontend
  + Ebben a mappában találhatóak a frontend összes fájlja.
* tests
  + Ebben a mappában találhatóak a program tesztjei.

### A fájlok:

* .gitignore -> ez szolgál a git repóba nem feltöltendő fájlokra, mappákra
* README.md -> Rövid leírás a git repórol.

## Backend

A backend mappa megnyitásakor ezt a fájlstruktúrát láthatjuk.



2. ábra A backend mappa struktúrája

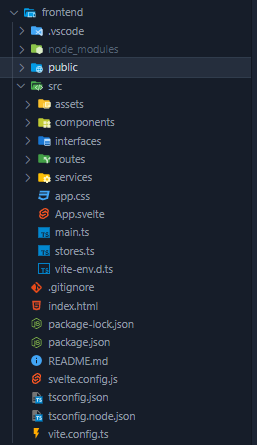
### A mappák:

* controllers
  + Ez a mappa tartalmazza a szerver által használt kontrollereket.
  + FileActions.js: Fájlfeltöltések endpointjai
  + mailingController.js: Levelezés endpontjai
  + sqlController.js: Az adatbázis lekérdezések endpointjai
  + usersController.js: Regisztrációs, bejelentkezési endpointok
* modules
  + Ez a mappa tartalmazza a szerver, és a kontrollerek által használt függvényeket, metódusokat.
  + logging.js: A szerver logolásának függvénye
  + tokenCheck.js: Authorizációs függvények.
* uploads
  + Ez a mappa tartalmazza a felhasználók által feltöltött tartalmakat.
* node\_modules
  + Ez a mappa tartalmazza a szerver által használt npm modulokat.

### A fájlok, és tartalmuk:

* .env: Szerver adatait tartalmazza
* .gitignore: git repóba nem feltöltendő adatok fájlja
* blacklist.json: kijelentkezéskor tárolt nem használandó tokenek.
* config.js: a szerver által használt adatok
* mailv2.ejs: A levelezés használt .html fájl
* server.js: A szerver

## Frontend

A frontend mappa megnyitásakor az alábbi mappastruktúrát láthatjuk

. ábra A frontend mappastruktúra

### A mappák

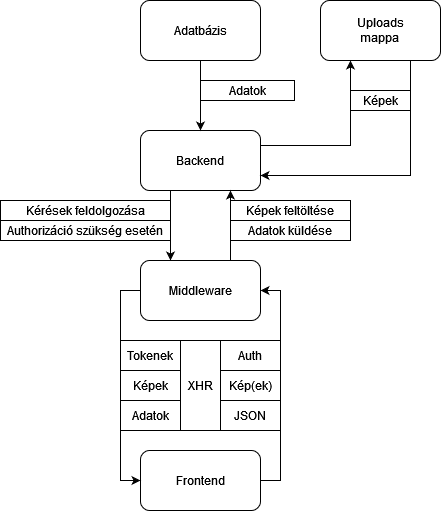
* .vscode: A VS Code-hoz segédfájlok
* node\_modules: A webalkalmazás moduljai
* public: A webalkalmazás statikus képei, videói, médiái
* src: A projekt forrásmappája
  + assets: Betűtípusok
  + components: Az oldalak fájlai
    - subcomponents: Az oldalak által használt alkomponensek fájlai
  + interfaces: Az alkalmazás által használt adatstruktúrák
  + routes: Az alkalmazás routing fájlai
  + services: Az alkalmazás backend lekérdezéseinek fájlai

### A fájlok, és tartalmuk

* .gitignore: A git repó által mellőzött fájlok
* index.html: Az alkalmazás törzsfájlja
* package-lock.json, package.json: Az alkalmazás által használt modulok listája
* README.md: Leírás a futtatásról,
* svelte.config.js, vite.config.ts: Az alkalmazás szerverének konfigurációs fájlja
* tsconfig.json, tsconfig.node.json: a TypeScript konfigurációs fájlja

# Az alkalmazás

## Kommunikáció



. ábra Az alkalmazás kommunikációjának diagramja

Az alkalmazás három részre bomlik: a backendre, frontendre, és az adatbázisra.

A frontend a backendnek 3 fajta adatot küldhet:

* JSON adatszerkezeteket
* Képeket
* Authorizációs tokeneket

A frontend és backend mindig XHTTPRequestekkel (XHR) kommunikál. A backend middleware-jei kezelik az authorizációkat, képfeltöltéseket, egyéb kéréseket.

## Adatbázis

Az adatbázis egy MySQL adatbázis. Importálás után a kozigi\_petak néven található meg az adatbázisok között. Az adatbázissal a szerver a „mysql” modult használja kommunikációhoz.

## Backend

A backend egy node.js express frameworkön futó API szerver. Az endpointokat a frontend hívja meg különböző kérések feldolgozásáért. Ezek lehetnek képek, vagy az adatbázisból lekért adatok. A szerver egy websocketet is tartalmaz. Ez kezeli az aukciós rendszert.

### Fájlfeltöltés

A FileActions.js fájlban található a szerver fájlfeltöltési funkciói.

A fájlok három szűrőn esnek át feltöltés előtt, amiket ebben a fájlban lehet módosítani

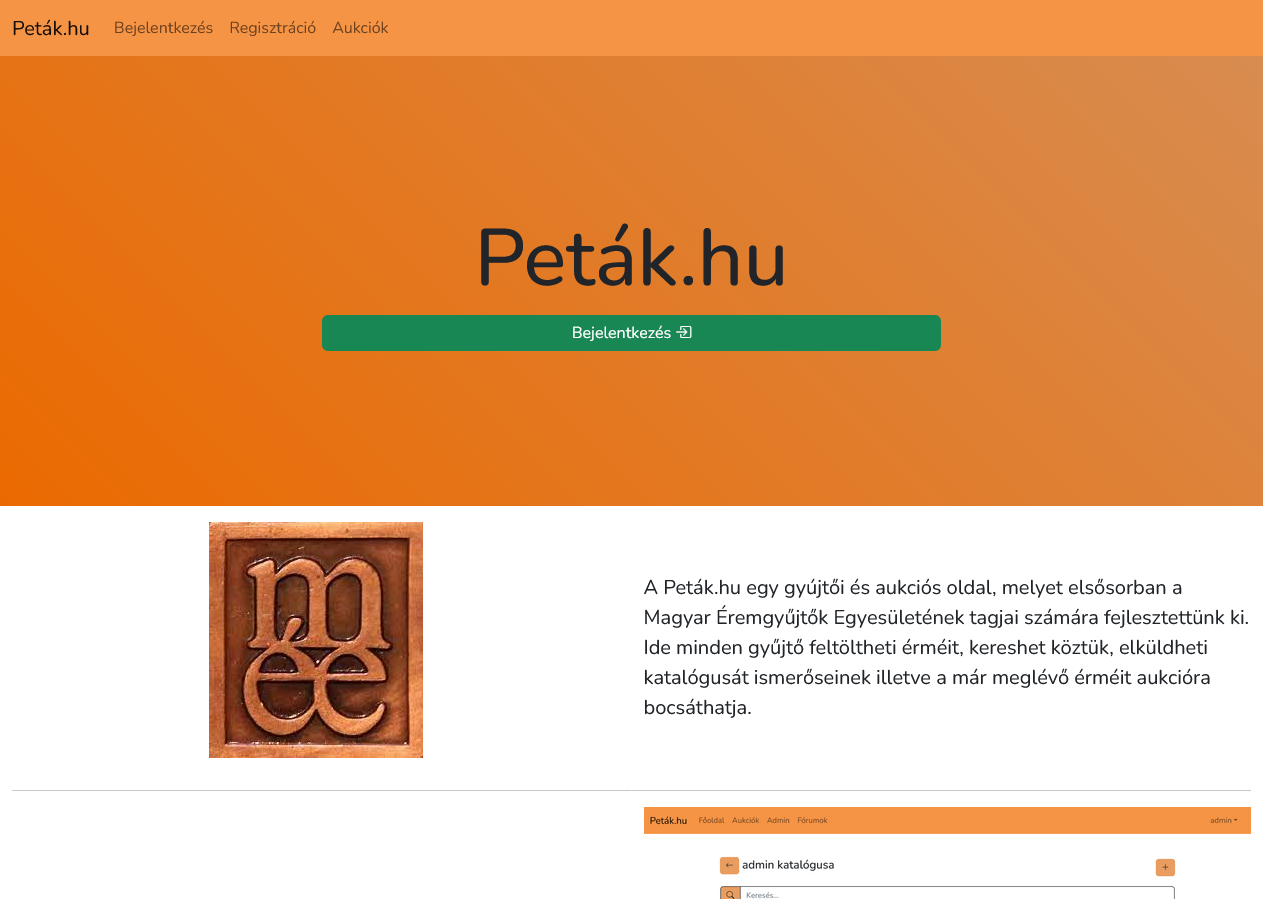
A szűrők:

* Fájlméret (jelenleg maximum 5MiB)
* Fájlfajta (csak képek)
* Fájlmennyiség (endpointtól függ, maximum 2 darab)

## Frontend

A frontend egy Svelte keretrendszeren futó webalkalmazás. Ez kommunikál a backenddel, illetve annak websocketjével. A rossz inputokat a frontend kezeli, így spórolva a backend teendőin. Az alkalmazás még a Bootstrap 5 CSS keretrendszert is használja.

Az oldal reszponzív. Legalább 275px széles képernyőméret szükséges



5. ábra A promóciós oldal nagy képernyőn



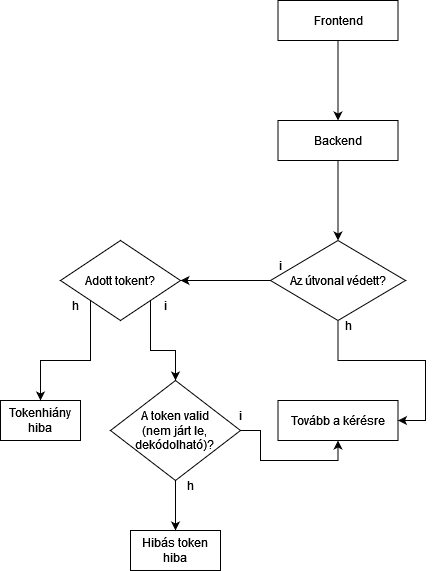
6. ábra A promóciós weboldal telefonokon

## Authorizáció

Némely útvonal „védett”, azaz nem bejelentkezett felhasználók, vagy nem jogosult felhasználók nem tekinthetnek meg némely lekérést megfelelő hitelesítés nélkül.

Ezek bármely adatmódosító, -törlő, hozzáadó endpointok, vagy „védettnek” számított tábla az SQL adatbázisban. A mindenki által elérhető táblák a tokenCheck.js fájl IsWhiteListedTable függvényben találhatóak.

Így a backend a JsonWebToken rendszer segítségével azonosítja az oldal felhasználóit.



. ábra Az authorizáció menete

A backendnek, ha tokent kell használnia egy lekérdezéshez, akkor három fajta választ tud adni.

* Rossz token (hibás formátum)
* Hiányos token
* Jó token

Jó token esetén a szerver tovább megy a lekérdezéssel, és annak az endpointnak megfelelően válaszol a frontendnek.

# Tesztek

Az oldal teszteléseit a tests mappában találhatóak, további teszteket ebben a mappában tárolunk.

A backend tesztjének Thunder-Client Collection-je a backend almappában találjuk. Ebben a tesztesetek is megtalálhatóak, és futtathatóak is.

# Továbbfejlesztés

## Frontend

* Új komponenseket az /frontend/src/components mappában ajánlott tárolni.
* Új alkomponenseket ugyanezen mappában a /subcomponents mappában ajánlott tárolni.
* Statikus képeket, illetve fájlokat a /frontend/public mappában kell tárolni

## Backend

* Új kontrollereket ajánlott a /backend/controllers mappában létrehozni
* Új modulokat, express middlewareket a /backend/modules mappában ajánlott létrehozni
* Újabb, szöveges titkosítandó adatokat ajánlott a .env fájlban tárolni.