

1. Úvod :

Název : My Home App

Popis : Modulární webová aplikace pro správu úkolů, poznámek, počasí, chytré domácnosti a více. Hlavním cílem je učení, trénink tvorby webových aplikací a prezentace implementovaného kódu.

Použité technologie:

Frontend :

- React,
- Vite,
- React Router (navigace mezi stránkami),
- Axios (volání backend API),
- Redux Toolkit (centralizovaná správa stavu),
- Redux Thunk (Async akce – pro komunikaci s API),
- Zod (validace dat na Frontend straně),
- React-Toastify (vizuální notifikace),
- Bootstrap + vlastní css + remix icon

Backend:

- Node.js + Express.js (server – REST API backend),
- MongoDB (database),
- Mongoose (Modelování dat a připojení k MongoDB),
- Joi (Validace vstupů),
- Nodemailer (odeslání emailové notifikace),
- JWT (jsonwebtoken – přihlášení a autorizace),
- express.json (parsování příchozích JSON souborů),
- cors (komunikace mezi různými doménami),
- dotenv (načítání proměných z .env souboru) ,
- bcrypt (hashování hesla),
- nodemon (vývojové prostředí),
- Tuay API (příprava pro připojení)

2. Frontend :

Popis:

Frontend aplikace je vytvořen pomocí Reactu a slouží jako uživatelské rozhraní pro komunikaci s backendem přes REST API. Zajišťuje:

- přihlašování a registraci uživatelů s validací vstupních údajů pomocí Zod,
- zobrazení a správu úkolů (To-Do modul) a poznámek (Notes modul),
- přepínání mezi různými sekcemi aplikace (např. Úkoly, Poznámky, správce uživatelů),
- správu stavu aplikace pomocí Redux Toolkit (včetně async komunikace pomocí Thunk),
- vizuální a systémové notifikace pomocí react-toastify a vlastního notificationSlice,
- responzivní a uživatelsky přívětivé rozhraní pomocí Bootstrap a vlastního CSS,
- zabezpečený přístup k chráněným stránkám pomocí ochrany rout a uložení JWT tokenu.

Struktúra:

client/

- node_modules/
- public/
- src/
 - components/ *(obecné komponenty pro celou aplikaci např. NotificationDisplay.jsx)*
 - layout/ *(komponenty layout například Sidebar.jsx)*
 - modules/ *(složky jednotlivých modulů, strukturované stejně)*
 - admin/ *(modul správa uživatelů)*
 - auth/ *(složka obsahující stránky pro přihlášení a pro registraci)*
 - home/ *(domovský modulu)*
 - notes/ *(poznámkový modul)*

- tasks/ (*úkolový modul – uvedena celá struktúra pro příklad*)
 - api/ (*klient pro komunikaci s backendem REST API*)
 - components/ (*jednotlivé komponenty modulu (TodoPage.jsx atd.)*)
 - hooks/ (*prázdná složka, převedeno na redux*)
 - pages/ (*hlavní komponenta TaskPage.jsx*)
 - styles/ (*css soubory pro komponenty ve složce*)
- styles/
- weather/
- pages/ (*komponenta LandingPage.jsx*)
- store/ (*Redux – store soubor, jednotlivé slice soubory ve složkách pro každý modul*)
- styles/ (*css soubor prokomponentu LandingPage.jsx*)
- validation/ schemas / (*schémata pro zod validaci*)
- AnimatedRoutes.jsx
- App.css
- App.jsx
- main.jsx
- store.zip/
- eslint.config.js
- index.html
- package.json
- package-lock.json
- README.md
- vite.config.js

Autentizace a přístupové stránky

Tato část aplikace zajišťuje správu přístupu k aplikaci – registraci nových uživatelů, přihlášení a přesměrování na hlavní obsah aplikace.

LandingPage.jsx

- Výchozí veřejná stránka aplikace.
- Obsahuje tlačítka pro přihlášení nebo registraci.
- Využívá React Router pro přesměrování (/login, /register).

LoginPage.jsx

Slouží k přihlášení uživatele do aplikace.

Obsahuje:

- vstupní pole pro jméno a heslo,
- validaci pomocí Zod,
- vizuální zpětnou vazbu (notifikace) pomocí React-Toastify.

V případě úspěšného přihlášení:

- uživatel je přesměrován do aplikace,
- obdrží JWT token pro autorizaci (uložený v localStorage).
- Využívá Redux pro správu přihlášení (authSlice.js).

RegistrationPage.jsx

Umožňuje vytvoření nového účtu.

Obsahuje:

- vstupní pole pro jméno, e-mail, heslo, pohlaví,
- validaci vstupních dat pomocí Zod (např. minimální délka hesla),
- notifikace pro úspěšnou/nezdařenou registraci.

Po odeslání žádosti:

- účet je vytvořen ve stavu „čekající na schválení“,
- uživateli přijde e-mail potvrzující odeslání žádosti (pomocí Nodemailer),
- superadmin poté účet schvaluje nebo odmítá v Admin panelu (správa uživatelů),
- Po schválení / zamítnutí registrace uživatele přijde e-mail o schválení / zamítnutí (pomocí Nodemailer),

Task modul :

Popis modulu :

Modul slouží pro zadávání a správu úkolů. Umožňuje:

- přidávat nové úkoly,
- vytvoření skupin,
- třídit úkoly do skupin,
- upravovat úkoly,
- označovat je jako dokončené,
- mazat úkoly i celé skupiny.

Skupiny úkolů jsou provázány – při odstranění skupiny se automaticky odstraní i všechny úkoly, které do ní patřily.

Každý úkol je reprezentován jako prvek seznamu a zobrazuje:

- text úkolu,
- datum vytvoření,
- prioritu (No Important, Important, High Important),
- a ovládací tlačítka (označit jako hotové, upravit, smazat).

Implementace :

- **Stavová logika:** pomocí Redux slice (tasksSlice.js)
- **Validace:** pomocí Zod (pro text úkolu, prioritu, skupinu)
- **Notifikace:** vizuální pomocí React-Toastify, stavové pomocí Redux (notificationSlice.js)

Použité Komponenty :

- **TodoPage.jsx** (Hlavní komponenta, obsshující komponenty TaskGroups.jsx, TaskList.jsx, TaskForm.jsx,)
 - **TaskGroups.jsx** (komponenta zobrazující seznam skupin úkolů a input pro vytvoření nové skupiny)
 - **TaskList.jsx** (Komponenta zobrazující seznam vytvořených úkolů)
 - **TaskForm.jsx** (komponenta zobrazující formulář pro přidání nového úkolu – popis úkolu, priorita, skupina. Zobrazeno pomocí Bootstrap modálu.)
 - **TaskDetail.jsx** (Komponenta zobrazující okno s EditTaskForm.jsx komponentou)
 - **EditTaskForm.jsx** (komponenta zobrazující vyplněný formulář daty od jednotlivých úkolů, možno měnit hodnoty a uložit)
-

Note modul :

Popis modulu :

Modul slouží pro zadávání a správu poznámek. Umožňuje:

- přidávat novou poznámku,
- označit poznámku barvou,
- vytvoření skupin,
- třídit poznámky do skupin, nebo jako poznámky pro vytvořené úkoly v Task modulu
- upravovat poznámky,
- mazat poznámky i celé skupiny.

Skupiny poznámek jsou provázány – při odstranění skupiny se automaticky odstraní i všechny poznámky, které do ní patřily.

Každý úkol je reprezentován jako prvek seznamu a zobrazuje:

- Nadpis poznámky,
- datum vytvoření,
- Text poznámky,
- Přiřazená barva,
- a ovládací tlačítka (upravit, smazat).

Implementace :

- **Stavová logika:** pomocí Redux slice (noteSlice.js)
- **Validace:** pomocí Zod (pro nadpis, text poznámky, barvu, skupinu nebo úkol)
- **Notifikace:** vizuální pomocí React-Toastify, stavové pomocí Redux (notificationSlice.js)

Použité Komponenty :

- **NotePage.jsx** (Hlavní komponenta, obsshující komponenty NoteGroups.jsx, NoteList.jsx, NoteForm.jsx,)
- **NoteGroups.jsx** (komponenta zobrazující seznam skupin poznámek a input pro vytvoření nové skupiny. Zobrazuje i seznam úkolů jako přepínací menu)
- **NoteList.jsx** (Komponenta zobrazující seznam vytvořených poznámek)
- **NoteForm.jsx** (komponenta zobrazující formulář pro přidání nové poznámky – nadpis poznámky, text poznámky, barva, skupina nebo úkol. Zobrazeno pomocí Bootstrap modálu.)
- **NoteDetail.jsx** (Komponenta zobrazující okno s EditNoteForm.jsx komponentou)
- **EditNoteForm.jsx** (komponenta zobrazující vyplněný formulář daty od jednotlivých poznámek, možno měnit hodnoty a uložit)

Admin modul :

Popis modulu :

Modul slouží pro správu uživatelů. Umožňuje:

- Schválit/zamítnout žádost o registraci nového uživatele,
- Blokovat účty uživatelů,
- Měnit role uživatelů (user, admin, superadmin),
- Udělovat/odebírat přístupy k jednotlivým modulům,
- Trvale odstranit uživatele i s jeho daty,

Každý uživatel je reprezentován jako prvek seznamu a zobrazuje:

- Jméno,
- Roli,
- a ovládací tlačítka (blokovat, upravit roli a přístupy, smazat).

Implementace :

- **Stavová logika:** pomocí Redux slice (adminSlice.js)
- **Notifikace:** vizuální pomocí React-Toastify, stavové pomocí Redux (notificationSlice.js)

Použité Komponenty :

- **AdminPanel.jsx** (Hlavní komponenta obsahuje, volbu potvrzení/zamítnutí registrace nového uživatele. Seznam všech uživatelů s možností blokace, změny role a přístupů nebo odstranění účtu)

SmarHome modul :

Popis modulu :

Prozatím není implementováno, nastaven pouze backend připojení na Tuya API.

3. Backend :

Popis:

Backend aplikace je postaven pomocí Node.js a Express.js a slouží jako REST API pro frontend. Zajišťuje:

- správu uživatelů (registrace, přihlášení, schvalování),
- správu úkolů a poznámek,
- komunikaci s databází MongoDB pomocí Mongoose,
- validaci vstupních dat pomocí Joi,
- odesílání e-mailových notifikací pomocí Nodemailer,
- autorizaci pomocí JWT tokenů.

Struktúra:

server/

- **controllers/** (*Obsluhuje logiku pro routy*)
- **middleware/** (*Middleware pro autentizaci, chyby atd.*)
- **models/** (*Mongoose modely (User, Task, Note, Group...)*)
- **routes/** (*API endpointy*)
- **services/** (*Pomocné služby (emailService.js)*)
- **utils/** (*logika pro note a task Groups controller (groupControllerUtils.js)*)
- **validation/** (*Joi schémata pro validaci dat*)
- **server.js** (*Vstupní bod aplikace a připojení k databázi*)

Autentizace:

- Uživatelská data jsou chráněna pomocí JWT tokenů.
- Přístup ke chráněným routám (např. /api/tasks) je řízen middlewarem authMiddleware.
- Hesla jsou ukládána zahashovaná pomocí bcrypt.

Validace vstupů:

- Vstupní data jsou validována pomocí knihovny Joi před uložením do databáze.

E-mailové notifikace:

- Odesílány přes nodemailer.

Příklady:

- odeslání registrace,
- schválení/zamítnutí registrace,
- oznámení o zablokování nebo smazání účtu,
- oznámení o změně role,
- oznámení o odstranění účtu.

API Routy:

Login and Registration:

POST /api/auth/register *(Registrace nového uživatele)*
POST /api/auth/login *(Přihlášení uživatele)*
GET /api/auth/me *(Získání profilu přihlášeného uživatele)*

Tasks:

GET /api/todo *(Získání úkolů podle skupiny a stavu)*
POST /api/todo *(Vytvoření nového úkolu)*
PUT /api/todo/:id *(Úprava úkolu)*
DELETE /api/todo/:id *(Smazání úkolu)*

TaskGroups:

GET /api/taskGroups *(Získání všech skupin úkolů)*
POST /api/taskGroups *(Vytvoření nové skupiny úkolů)*
DELETE /api/taskGroups/:id *(Smazání skupiny úkolů)*

Notes:

GET	/api/notes	(Získání všech poznámek)
POST	/api/notes	(Vytvoření nové poznámky)
PUT	/api/notes/:id	(Úprava poznámky)
DELETE	/api/notes/:id	(Smazání poznámky)

NoteGroups:

GET	/api/noteGroups	(Získání skupin poznámek)
POST	/api/noteGroups	(Vytvoření nové skupiny poznámek)
DELETE	/api/noteGroups/:id	(Smazání skupiny poznámek)

Admin:

GET	/api/admin/pending-registrations	(Získání žádostí o registraci)
PUT	/api/admin/approve/:id	(Schválení uživatele)
DELETE	/api/admin/reject/:id	(Zamítnutí žádosti o registraci)
GET	/api/admin/users	(Získání všech uživatelů)
PUT	/api/admin/users/:id/role	(Změna role uživatele)
PUT	/api/admin/users/:id/access	(Změna přístupů k modulům)
PUT	/api/admin/permissions/:id	(Správa oprávnění uživatele)
PUT	/api/admin/users/:id/status	(Aktivace / blokace účtu)
DELETE	/api/admin/users/:id	(Smazání uživatele)

4. Plánované vylepšení :

Obecné:

- 1) Více upravit kód do obecné roviny. Vytvořit obecné komponenty pro všechny opakované elementy např. Inputs, selects, buttons.
- 2) Opravit logiku rolí, aktuálně - user, admin, superadmin. Role admin nemá rozlišná práva oproti user.
- 3) Projít a revidovat zod validaci, přidat restrikce např. pro maximální počet znaků.

Plán na úpravu kompetencí :

- User => pouze moduly,
 - Admin => moduly + schvalování nových registrací,
 - Superadmin => moduly + schvalování nových registrací + správa současných uživatelů.
- 4) Vymodelovat UML diagramy v rámci procvičování a pro popis aplikace jak v rovině náčrtu tak v rovině návrhu.
 - 5) Upravit zvhled aplikace, více responzivní a uživatelsky příjemný.
 - 6) Doplnit Landing Page, prozatím prázdná. Implementovat co aplikace nabízí.

Task modul:

- 7) Upravit spojení úkolů a úkolových poznámek, upravit aby po odstranění úkolu se odstranila i připojená poznámka.
- 8) Přidat možnost vytvořit podúkoly k jednotlivým úkolům. Jak při vytváření úkolu tak při editaci.
- 9) Přidat možnost sdílet úkol i s jiným uživatelem.

Home modul:

- 10) Přidat výtah z každého modulu. Rychlý přístup k požadovaným informacím.

5. Plánované nové moduly :

1) SmartHome modul :

- Dokončit/Vytvořit UI pro chytrou domácnost, vyhledání a připojení chytrého zařízení (čidla, světla, teploměr atd.) čtení dat, ovládání stavů, nastavení scénářů.
- backend prozatím spojen s Tuya API chytré zařízení, po hlubším zkoumání zřejmě není vhodné. Nutno porovnat ostatní výrobce a fungování lokálních nebo cloudových zařízení.

2) Modul Kalendář :

- Vytvořit jednoduchý UI pro kalendář s možností vytvářet eventy v určitém datu a čase.
- Propojení s modulem úkolů, možnost naplánovat plnění úkolu na konkrétní den a čas/časový interval.

3) Modul Počasí :

- Vytvořit UI pro data o počasí, pomocí OpenWeatherMap API nebo WeatherAPI.com kombinovat s OpenAI API.
- Zadání lokality, detailní data o počasí.
- Upozornění na stav v budoucnu, možnost plánovat výlet (pokud budou ideální podmínky na základně ročního období, letní výlety = more, nebo zimní = lyžování)
- Propojení s modulem kalendáře, viditelně ukázaný termín, kdy je nejlepší vyrazit na výlet.

4) Investiční modul :

- Vytvořit UI investičního asistenta kombinací Alpha Vantage API a OpenAI API.
- Doporučení firem podle zájmů uživatele.
- Nacenění reálné ceny akcie, poradenství kdy koupit a kdy prodat dle nastavení strategie podle uživatele.
- Doporučování nových společností dle preference uživatele.