

Střední odborná škola a střední odborné učiliště,  
Hořovice  
Palackého náměstí 100, 268 01 Hořovice

---



## **Webová aplikace pro interpretaci vybraných astronomických dat**

Adam Huml

2022/2023

Studijní obor: Informatika v ekonomice

Třída: 4.A

Rok obhajoby: 2023

Vedoucí práce: Ing. Vladimír Kebert, CSc.

## Zadávací list

## Webová aplikace pro interpretaci vybraných astronomických dat

### Anotace

Cílem aplikace je sdružení informací o astronomických tělesech do jedné databáze a jejich následná interpretace uživateli. Administrátor má možnost prostřednictvím aplikace vkládat data do databáze, měnit je, či mazat. Uživatel má poté možnost si tato data číst, diskutovat o nich na diskusním fóru, nebo případně využít API k čerpání těchto dat např. pro jeho vlastní aplikaci.

### Klíčová slova

PHP, Python, Mysql, Uvicorn, FastAPI, 3D modely, vizualizace, interpretace , webová aplikace,

## Web application for interpretation of selected astronomical data

### Annotation

The aim of this application is to combine information about astronomical bodies into one database and their subsequent interpretation by the user. The administrator has the option to insert, change or delete data in the database using the application. The user, on the other hand, has the option to read this data, discuss it on the discussion forum, or possibly use the API to draw this data, for example, for his own application.

### Keywords

PHP, Python, Mysql, Uvicorn, FastAPI, 3D models, visualisation, interpretation, web application

### ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že jsem závěrečnou práci na téma: „Webová aplikace pro interpretaci vybraných astronomických dat“ vypracoval samostatně na základě citované literatury a na základě vlastního výzkumu a experimentování. Prohlašuji, že celá práce včetně experimentálních výsledků, může být dále využívána pro potřeby Střední odborné školy a Středního odborného učiliště Hořovice.

V Hořovicích, dne 1. března 2023 .....

**Poděkování**

# Obsah

<b>1. Úvod</b>	9
<b>2. Registrace</b>	10
2.1 Validace registračních údajů	11
2.2 Design registrace	12
2.3 Přihlášení	13
2.4 Design přihlášení	13
<b>3. Bezpečnost stránky</b>	14
3.1 Ochrana před „Path traversal“ útokem	14
3.2 Ochrana před „SQL injection“ útokem	16
3.3 Ochrana před „session hijacking“ („ukradení sessiony“)	17
3.3.1 Využití HTTPS	17
3.3.2 Zabránění XSS – „Cross site scripting“	17
3.3.3 Změna klíče sessiony	17
<b>4. Struktura projektu</b>	18
4.1 Adresářová struktura	18
4.2 Struktura MySQL databáze	18
<b>5. Administrace</b>	19
5.1 Ověření, zdali je uživatel administrátorem	19
5.2 Design administrace	20
5.3 Funkce administrace	21
5.3.1 Funkce správy uživatelů	21
5.3.2 Funkce správy API klíčů	21
<b>6. Správa účtu uživatele</b>	22
6.1 Generování API klíčů	23
<b>7. API</b>	24
7.1 Třída DBHandler	24
7.2 Nastavení resources API – api.py	25
7.3 Funkce check_api_key(api_key)	25
7.4 Oddělení API od aplikace	26
7.5 Výstup z API	26
<b>8. Astronomické objekty</b>	27
8.1 Vyhledávač	27
8.2 Planety sluneční soustavy	28

8.3	Satelity .....	29
8.4	Hvězdy .....	30
8.5	Exoplanety .....	31
9.	Manipulace s daty .....	32
9.1	Vkládání nových objektů .....	32
9.2	Úprava objektů .....	33
9.3	Mazání objektů .....	34
10.	Uživatelské diskusní fórum .....	35
10.1	Implementace fóra do aplikace .....	36
11.	Závěr .....	37
12.	Seznam použitých zdrojů .....	38
12.1	Literární řešerše .....	38
12.2	Použito při tvorbě práce .....	39
12.3	Použito při plnění práce daty .....	39
12.4	3D modely .....	39

## Seznam ukázek kódu

Ukázka kódu 1 – ověření shody hesel .....	11
Ukázka kódu 2 – ověření shody hesel .....	11
Ukázka kódu 3 – validace e-mailové adresy .....	11
Ukázka kódu 4 – ověření duplikátů .....	11
Ukázka kódu 5 – form-control - Bootstrap .....	12
Ukázka kódu 6 – soubor querystring_h.php .....	15
Ukázka kódu 7 – escapování znaků .....	16
Ukázka kódu 8 - funkce generující UUID 4 API klíče <sup>[8]</sup> .....	23
Ukázka kódu 9 - třída DBHandler - inicializace třídy DBHandler pomocí proměnných prostředí .....	24
Ukázka kódu 10 - třída DBHandler - funkce gather_data umožňující výběr dat z databáze .....	25
Ukázka kódu 11 - api.py - ukázka funkce check_api_key .....	26
Ukázka kódu 12 - Uživatelské diskusní fórum - ukázka implementace přidání nového uživatele do fóra .....	36

## Seznam tabulek

Tabulka 1 - MySQL sloupec role.....	19
Tabulka 2 - MySQL sloupec password_reset (reset hesla).....	21

## Seznam obrázků

Obrázek 1 – ukázka designu registrace.....	12
Obrázek 2 - Pravá část menu po přihlášení administrátora s už. jm. Xehos .....	13
Obrázek 3 – ukázka designu přihlášení.....	13
Obrázek 4 - ukázka designu administrace - uživatelé .....	20
Obrázek 5 - ukázka designu administrace - API klíče .....	20
Obrázek 6 - administrace - ukázka úpravy uživatele.....	21
Obrázek 7 - účet - ukázka bez vygenerovaného API klíče .....	22
Obrázek 8 - účet - ukázka s vygenerovaným API klíčem .....	22
Obrázek 9 - ukázka nastavení NGINX reverzní proxy pro API .....	26
Obrázek 10 - objekty - design vyhledávače .....	27
Obrázek 11 - Objekty - design záznamu planety sluneční soustavy .....	28
Obrázek 12 - Objekty - design záznamu satelitu.....	29
Obrázek 13 - Objekty - design záznamu hvězdy .....	30
Obrázek 14 - Objekty - design záznamu exoplanety.....	31
Obrázek 15 - Manipulace s daty - ukázka tlačítka pro vložení nového objektu (planety) .....	32
Obrázek 16 - Manipulace s daty - ukázka designu formuláře pro vložení nového objektu (planety) ...	32
Obrázek 17 - Úprava objektů - ukázka tlačítka pro úpravu objektu (planety) .....	33
Obrázek 18 - Úprava objektů - design úpravy objektu (planeta sl. soustavy - Merkur) .....	33
Obrázek 19 - Mazání objektů - ukázka tlačítka pro smazání objektu (planety) .....	34
Obrázek 20 - Mazání objektů - ukázka potvrzení smazání objektu .....	34
Obrázek 21 - Uživatelské diskusní fórum - Ukázka fóra na platformě phpBB® po instalaci.....	35
Obrázek 22 - Uživatelské diskusní fórum - deaktivace registrací v phpBB® .....	36
Obrázek 23 - Uživatelské diskusní fórum - ukázka přihlášení do fóra phpBB® .....	36
Obrázek 24 - design domovské stránky aplikace .....	37
Obrázek 25 - naplnění aplikace daty z modelu umělé inteligence GPT-3 společnosti OpenAI Inc. ....	37



# 1. Úvod

Téma této maturitní práce si student zvolil, jelikož se ve volném čase zajímá o astronomii a chtěl si rozvinout své znalosti v oblasti SQL databáze a práce s ní.

Celý projekt student zpracovával přibližně rok. Při výběru nástrojů zvažoval různé způsoby, kterými by se práce dala zhotovit.

Nakonec se rozhodl pro následující:

1. Databáze: MariaDB
2. Programovací jazyk pro stránky: PHP (objektově)
3. Stránky (design): JS+HTML5+CSS3+Bootstrap
4. Grafika – CorelDraw
5. FTP server - FileZilla
6. API – Python (modul FastAPI)
7. Textový editor pro psaní kódu: Sublime Text
8. Uživatelské diskusní fórum: phpBB®
9. Server: částečně vyvíjel na XAMPP, produkční verze běží na Apache2 (Linux)

Stránky jsou postaveny na Bootstrapu v. 4.6.2, který zároveň zajišťuje jejich responzivitu.

Dále bylo třeba zvážit, která vybraná data budou na stránce zveřejněna. Student vytvořil na webu 4 stránky na kterých se nachází vybrané objekty – Planety sluneční soustavy, Satelity (umělé i přirozené), Hvězdy, Exoplanety.

Co se pak týče jednotlivých dat, tak pro účel práce vybral ty nejdůležitější parametry těles, jako je jejich název, hmotnost, perioda orbity planety/slunce apod. Zároveň je také možné interpretovat 3D model každého z těles, což uživateli umožňuje prohlédnout si jej ze všech stran.

Jako zdroj informací student využil různé internetové stránky viz stránka citací.

Velice důležitou částí webové aplikace je také bezpečnost a ochrana před narušením. Součástí této práce byla například ochrana před „Path traversal“ útokem, nebo „SQL injection“ útokem.

K nahrávání 3D modelů používá admin Filezilla FTP server. K tomuto způsobu jsem bylo přistoupeno, jelikož jsou 3D modely mnohdy datově objemné a jejich nahrávání přes POST request je tudíž velmi neefektivní. Byla by pochopitelně možnost použít PHP framework s lepším handlingem dané problematiky, to již ale zasahuje mimo rozsah této práce.

## 2. Registrace

Každý uživatel, který chce na stránce provádět akce, kromě pasivního zobrazení listu astronomických těles, musí být registrován. Registrace se provádí na základě následujících údajů:

- Uživatelské jméno
- E-mail
- Jméno
- Příjmení
- Heslo (2x pro kontrolu)
- Místo narození
- Datum narození

Místo narození může uživatel zvolit pomocí 3 listů – států, krajů a měst. K tomuto jsem využil tzv. AJAX pro requesty databáze. Ten umožňuje číst z databáze na základě možnosti vybrané uživatelem, aniž by se webová stránka musela obnovit.

S AJAXem student žádné předchozí zkušenosti neměl, každopádně se mu jeho využití velmi osvědčilo. Lze téměř říct, že je možné jej použít kdekoliv, kde programátor vyžaduje dynamicky obnovovanou stránku v PHP, ovšem nechce, aby se celá obnovovala.

Vybírání místa narození tedy funguje tak, že uživatel vybere stát, poté se automaticky pošle request na PHP stránku, která vrátí seznam krajů/provincií daného státu a nimi naplní list pod státem. Tímhle způsobem se uživatel dostane až ke svému rodnému městu.

Heslo je do databáze uloženo jako hash pomocí ARGON2I algoritmu.

Celý formulář je kvůli bezpečnosti odesílán na server metodou POST (chráněnou escapováním znaků viz kapitola 4., odkud se poté vytvoří nový uživatel v databázi. Před tím prochází registrační údaje několika kontrolami. (jejich popis se nachází na další stránce) V případě že nejsou registrační údaje v pořádku – je to uživateli oznámeno chybovou hláškou.

Registrace je chráněna proti zneužití CAPTCHOU společnosti Google. Její algoritmus rozpozná, zdali se chce registrovat člověk, či nějaký automatický skript a případně registraci zakáže.

Po registraci je uživatel přesměrován na stránku přihlášení.

## 2.1 Validace registračních údajů

Registrační údaje prochází hned několika validačními funkcemi.

### 1.) Ověření, že uživatel opravdu vyplnil veškeré údaje

Ačkoliv tagy input polí pro registrační údaje disponují „required“ atributem, je lepší ověřovat skutečnost, že uživatel vyplnil údaje, i na serverové straně.

K tomu byla využita metoda **isset**, ta ověří, jestli daná proměnná existuje.

```
if(!isset($postdata["city"])){  
    return "Nezadali jste město!";  
}
```

*Ukázka kódu 1 – ověření shody hesel*

### 2.) Ověření, zdali se zadaná hesla shodují

Spočívá v porovnání obou zadaných hesel.

```
if($password!=$password_check){  
    return "Zadaná hesla se neshodují!";  
}
```

*Ukázka kódu 2 – ověření shody hesel*

### 3.) Ověření, zdali je e-mailová adresa zadána v korektním formátu

K ověření byl využit PHP flag pro metodu

filter\_var - **FILTER\_VALIDATE\_EMAIL**

```
function validateEmail($email){  
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
        return false;  
    }else{  
        return true;  
    }  
}
```

*Ukázka kódu 3 – validace e-mailové adresy*

### 4.) Ověření duplikátů uživatelských jmen a e-mailových adres

Ověření pomocí dotazů do databáze.

```
function verifyMailDuplicates($mail){  
    global $conn;  
    $sql = "SELECT COUNT(*) FROM astronnet_users WHERE mail='$mail' LIMIT 1";  
    $dataraw = Db::querySingle($sql);  
    if ($dataraw) {  
        return false;  
    }else{  
        return true;  
    }  
}
```

*Ukázka kódu 4 – ověření duplikátů*

## 2.2 Design registrace

The screenshot shows a registration form on the AstroNet website. The header includes navigation links: Domů, Astronomické objekty, API, O projektu, and a login link Přihlásit se. The form is titled "Registrace:" and contains several input fields arranged in a grid. The first row has "Uživatelské jméno" and "E-mail". The second row has "Jméno" and "Příjmení". The third row has "Heslo" and "Heslo pro kontrolu". Below these is a dropdown for "Prosím vyberte pohlaví". Then, "Místo narození:" is followed by a dropdown for "Czechia" and another for "Prosím vyberte kraj/stát/provincii". Below that is "Datum narození:" with a date picker showing "21.01.2023". A "Registrovat se" button is at the bottom of the form. At the very bottom, there is a link "Máte již účet? Prosím přihlašte se".

Obrázek 1 – ukázka designu registrace

Pro registraci byla využita především bootstrapová třída „form-control“<sup>[1]</sup>, pomocí které byla vytvořena responsivní input textová pole. Tato pole jsou na stránce rozvržena pomocí gridu.

```
<form action="" method="post">
  <div class="row justify-content-center">
    <div class="col-xs-6">
      <input type="text" name="username" tabindex="1" class="form-control form-item mt-2" placeholder="Uživatelské jméno" required>
      <input type="text" name="name" tabindex="3" class="form-control form-item mt-2" placeholder="Jméno" required>
    </div>
    <div class="col-xs-6">
      <input type="password" name="password" tabindex="5" class="form-control form-item mt-2" placeholder="Heslo" required>
      <input type="email" name="mail" tabindex="2" class="form-control form-item mt-2 ml-1" placeholder="E-mail" required>
      <input type="text" name="surname" tabindex="4" class="form-control form-item mt-2 ml-1" placeholder="Příjmení" required>
      <input type="password" name="password_check" tabindex="6" class="form-control form-item mt-2 ml-1" placeholder="Heslo pro kontrolu" required>
    </div>
  </div>
```

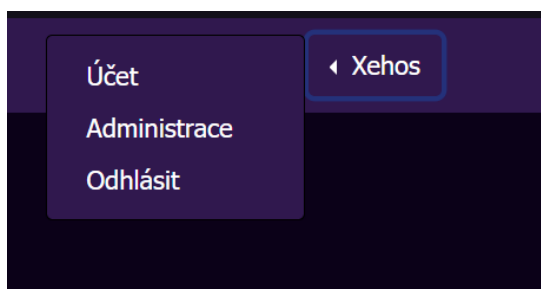
Ukázka kódu 5 – form-control - Bootstrap

<sup>1</sup>Form controls. Bootstrap [online]. 2022 [cit. 2023-01-21]. Dostupné z: <https://getbootstrap.com/docs/4.6/components/forms/#form-controls>

## 2.3 Přihlášení

Při přihlašování je uživatelem zadaná e-mailová adresa vyhledána v databázi. Pokud je nalezena, je následně hash hesla v databázi porovnán s hashem hesla, které uživatel zadal k přihlášení.

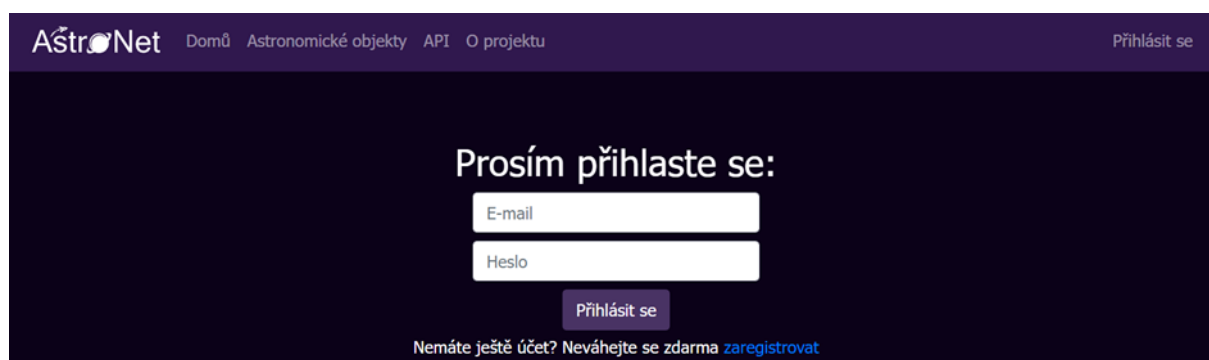
Pokud hashe hesel sedí, je uživatel přihlášen a jeho údaje zapsány do SESSION super-proměnné. Pravá část menu je poupravena tak, aby přihlášenému uživateli umožnila odhlášení, nebo úpravu účtu, případně adminovi přístup do administrace.



Obrázek 2 - Pravá část menu po přihlášení administrátora s už. jm. Xehos

## 2.4 Design přihlášení

Design je zachován stejný, jako v případě registrace.



Obrázek 3 – ukázka designu přihlášení

### 3. Bezpečnost stránky

Vzhledem k předpokladu umístění stránky na internet, musíme předpokládat pokusy o její potenciální napadení/zneužití.

Ve svojí práci se student různými způsoby snažil předcházet různým druhům hackerských útoků. Mezi ně patří:

#### 3.1 Ochrana před „Path traversal“ útokem

Path traversal útok využívá zranitelnosti webu v závislosti na nedostatečném oddělení aplikace od zbytku serveru. Útočník se tak například pomocí znaků pro listování v adresářích (../) může dostat mimo složku aplikace, kde může dále číst citlivá data serveru.

V mojí aplikaci by teoreticky k problému s Path traversal útokem mohlo dojít v místě, kde jsou querystringem includované jednotlivé stránky ze složky „pages“. Kdybych metodu include patřičně neošetřil, mohl by útočník do webové stránky dostat libovolný soubor ze serveru (ke kterému by pochopitelně měl přístup serverový uživatel, na jehož účtu PHP server běží)

Příklad:

Kdyby se útočník pokusil o přístup k takové stránce například s následujícími atributy:

`http://server:port/zranitelna-stranka.php?stranka=../../../../../../etc/passwd`

mohl by v závislosti na nastavení Linuxového serveru získat soubor `/etc/passwd`

Odpověď serveru by vypadala zhruba takto:

```
HTTP/1.0 200 OK
Content-Type: text/html
Server: Apache

root:fi3sED95ibqR6:0:1:System Operator:/:/bin/sh
daemon*:1:1:/:/tmp:
phpuser:f8fk3j10If31.:182:100:Developer:/home/users/phpuser/:/bin/csh
```

Proti path traversal útoku se student rozhodl stránku bránit pomocí kontroly obsahu querystringu.

Skript `querystring_h.php` mi umožňuje kontrolovat hodnoty atributu `page` a includovat obsah pouze v případě, že se stránka nachází v poli povolených stránek.

Zároveň je pomocí funkcí `htmlspecialchars` a `stripslashes` zabráněno vkládání speciálních znaků, jako jsou zpětná lomítka, což také znemožňuje útočníkovi provést tento druh útoku viz kapitola [3.2.](#)

```

<?php
/*

Used to redirect pages via appropriate querystring using "include"
IMPORTANT NO OTHER PAGES CAN WORK WITHOUT ADDITION TO THIS ARRAY!

*/
$pages = array("domu","API","o_projektu", "objekty",
    "login","register", "account",
    "edit", "administration", "resetpass");

if(isset($_GET['page'])){
$page = htmlspecialchars(stripslashes($_GET["page"]));
if(in_array($page,$pages)){
    include "pages/$page"."php";
    echo "
    <script>
        menu_selected('$page');
    </script>";
}
else{
    header("Location: index.php?page=domu");
}

}
else{
    header("Location: index.php?page=domu");
}

?>

```

*Ukázka kódu 6 – soubor querystring\_h.php*

Pokud by stránka nebyla v poli \$pages → systém nepovolí její vložení.

### 3.2 Ochrana před „SQL injection“ útokem

SQL injection využívá zranitelnosti neošetřených příkazů do databáze. Nejzranitelnější jsou tedy místa aplikace, kde uživatel zadává předem neurčené údaje v podobě textu (řetězce) a tento řetězec je následně zapsán do databáze bez předchozí validace.

V takovém případě existují různé příkazy, kterými může útočník získat z databáze citlivé informace, aniž by o tom věděli administrátoři.

Příklad<sup>[2]</sup>:

```
$sql="SELECT * FROM uzivatele WHERE jmeno = '$_POST['jmeno']'";  
Db::queryAll($sql);
```

Pokud by uživatel vyplnil do vstupního pole pro jméno následující řetězec:

```
a' or 'b'='b
```

Výsledný řetězec by vypadal takto:

```
$sql="SELECT * FROM uzivatele WHERE jmeno = 'a' or 'b'='b'";
```

což může zapříčinit přemostění autentizační procedury, protože 'b' = 'b' je vždy pravda, tudíž klientská (zobrazovací) vrstva aplikace vypíše všechny uživatele (nejen s jménem 'a'), pokud se jedná o stránku záznamů, tj. ne stránku vlastností jednoho záznamu se jménem 'a'.

K ochraně proti SQL injection student využil metody **htmlspecialchars** a **stripslashes**, které umožňují odstranit z řetězců potenciálně nebezpečné znaky – „escapovat“ je.

```
$mail = htmlspecialchars(stripslashes($_POST["mail"]));  
$password = htmlspecialchars(stripslashes($_POST["password"]));
```

*Ukázka kódu 7 – escapování znaků*

Jinou možností by bylo například využití REGEX (regulérního výrazu) a metody **preg\_match()**.

---

<sup>2</sup>SQL Injection. Wikipedia [online]. 2010-05-26 [cit. 2023-01-22]. Dostupné z: [https://en.wikipedia.org/wiki/SQL\\_injection?oldid=364265627](https://en.wikipedia.org/wiki/SQL_injection?oldid=364265627)



### 3.3 Ochrana před „session hijacking“ („ukradení sessiony)

Ve webové aplikaci, která využívá ke svému fungování super-proměnnou `$_SESSION` je nutné myslet na možnost jejího potenciálního ukradení útočníkem.

Bránit se tomuto útoku je možné několika následujícími způsoby:

#### 3.3.1 Využití HTTPS

HTTPS protokol oproti jeho předchůdci HTTP využívá v rámci přenosu dat od uživatele a zpět k němu šifrování (**SSL/TLS**) a **snižuje tak riziko zneužití osobních údajů**, záměny obsahu či odposlech online komunikace. Zároveň útočníkům zabraňuje dostat se k ID sessiony a tudíž ji „ukrást“. <sup>[3]</sup>

#### 3.3.2 Zabránění XSS – „Cross site scripting“

Cross site scripting umožňuje útočníkům dosazovat na stránku vlastní JavaScriptové soubory, nimiž mohou stránku ovládat a teoreticky získat ID sessiony. Blokace XSS na úrovni webového serveru je jedním z dalších způsobů, jak se bránit session hijackingu. <sup>[4]</sup>

#### 3.3.3 Změna klíče sessiony

Další účinnou metodou je také změnit klíč sessiony, jakmile se uživatel přihlásí. To způsobí, že ID sessiony, které útočník získá, je mu úplně k ničemu, jelikož klíč je nyní jiný. <sup>[4]</sup>

Student za účelem ochrany své aplikace využil HTTPS certifikátu **Let's Encrypt**, který zdarma poskytuje ochranu pomocí šifrování přenášených dat.

**Let's Encrypt** je certifikační autorita. Za pomoci automatizovaného procesu, navrženého tak, aby odstranil složitý proces manuální tvorby, ověřování, podepisování, instalace a obnovování certifikátu. Zdarma poskytuje doménově ověřené certifikáty typu **X.509** pro šifrování protokolu TLS. <sup>[5]</sup>

---

<sup>3</sup>KOĐOUSKOVÁ, Barbora. *HTTPS V KOSTCE: CO TO JE, JAK FUNGUJE A JAK NA NĚJ PŘEJÍT*. Rascasone [online]. 2021, 2021-11-17 [cit. 2023-01-22]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-https-http-ssl-tls>

<sup>4</sup>BAIG, Anas. *What is Session Hijacking and How Do You Prevent It?*. GlobalSign Blog [online]. 2021, 2021-07-09 [cit. 2023-01-22]. Dostupné z: <https://www.globalsign.com/en/blog/session-hijacking-and-how-to-prevent-it>

<sup>5</sup>MLEJNEK, Pavel, MAKEŠ, Václav, Jan KOVÁŘ BK a Milan KERŠLÁGER, ed. Let's Encrypt. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023, 2015-09-26 [cit. 2023-03-06]. Dostupné z: [https://cs.wikipedia.org/wiki/Let%27s\\_Encrypt](https://cs.wikipedia.org/wiki/Let%27s_Encrypt)

## 4. Struktura projektu

### 4.1 Adresářová struktura

Při tvorbě rozsáhlého webového projektu je důležité se předem zamýšlet nad jeho celkovou adresářovou strukturou. Ta se pochopitelně v průběhu vývoje mění, avšak základ zůstává stejný.

V rámci projektu byly vytvořeny následující adresáře:

- 3d – obsahuje 3D objekty
- backend – obsahuje Python backend pro práci s API
- classes – obsahuje třídy
- grafika – obsahuje grafické soubory
- img – obsahuje obrázky
- js – obsahuje JavaScriptové soubory
- pages – obsahuje jednotlivé stránky
- scripts – obsahuje krátké PHP skripty
- styles – obsahuje CSS styly

### 4.2 Struktura MySQL databáze

Součástí aplikace je MySQL (MariaDB) databáze s následujícími tabulkami:

Tabulka	Význam
astronet_api_help	Nápověda k API zdrojům
astronet_api_keys	API klíče
astronet_users	Uživatelé
astronet_ssplanets	Planety sluneční soustavy
astronet_satellites	Satelity
astronet_stars	Hvězdy
astronet_exoplanets	Exoplanety
astronet_countries	Země
astronet_states	Státy
astronet_cities	Města

- Žlutá = API
- Modrá = správa uživatelů
- Zelená = data aplikace
- Béžová = AJAX - výběr místa narození

## 5. Administrace

Webovou administrací se myslí správcovské rozhraní aplikace dostupné přes webový prohlížeč. V administraci typicky uživatelé na základě svých rolí mohou přistupovat k různým funkcím aplikace, jako je například správa obsahu, uživatelů nebo dalších funkcí.<sup>[6]</sup>

Konkrétním druhem administrace využitě v této práci je CMS (content management system). V tomto případě bývá administrace použita pro plnění dat/obsahu do aplikace.<sup>[6]</sup>

Ke stránce Administrace mají přístup pouze administrátoři webu.

Ti zde mají přístup ke dvěma podmenu

1. **Uživatelé** – zde mají možnost upravovat uživatele, mazat je, nebo resetovat jejich hesla
2. **API klíče** – zde mohou vidět API klíče jednotlivých uživatelů, případně je blokovat

Po resetu hesla je uživatel při dalším přihlášení vyzván k tomu, aby si vytvořil heslo nové.

### 5.1 Ověření, zdali je uživatel administrátorem

Ověření probíhá na základě příznaku „role“ uloženého v MySQL

Pokud je tento příznak roven 1 – jedná se o administrátora, pokud 0 – jedná se o běžného uživatele.

V databázi je sloupec role řešen následovně:

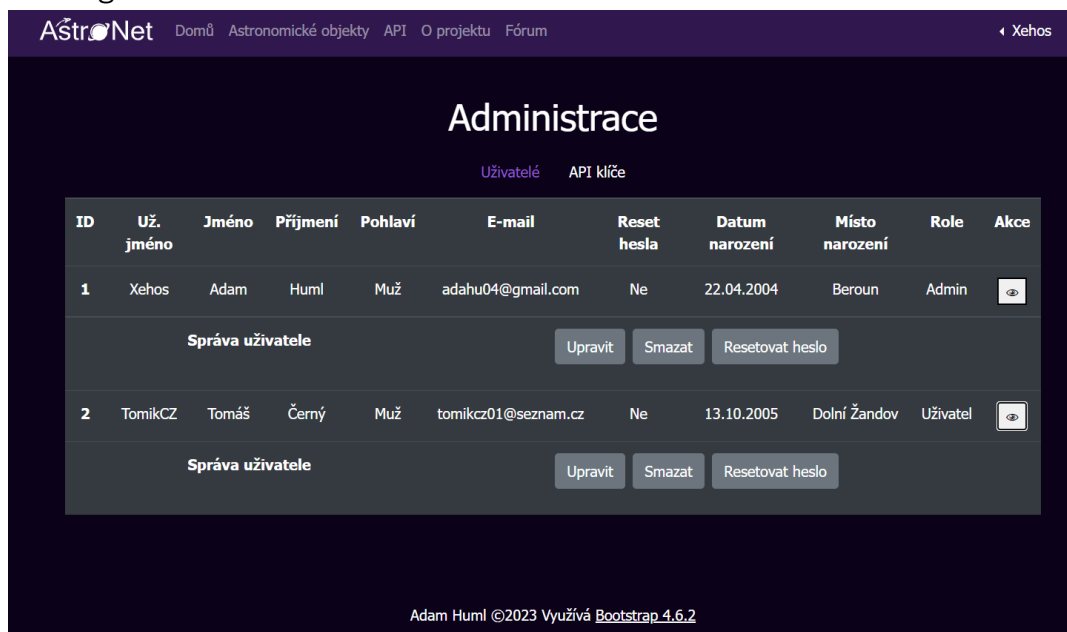
Název	Typ	Nulový	Výchozí
role	int(11)	Ne	0

*Tabulka 1 - MySQL sloupec role*

---

<sup>6</sup>Webová Administrace: Co to znamená "Webová Administrace"? In: *DAMI development* [online]. Česká republika: DAMI development [cit. 2023-03-05]. Dostupné z: <https://www.damidev.com/slovník/webova-administrace>

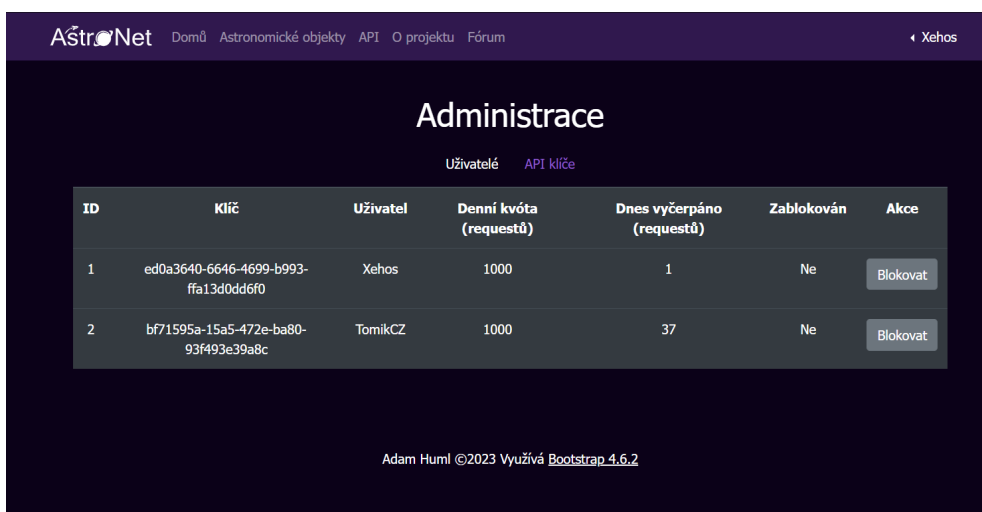
## 5.2 Design administrace



Obrázek 4 - ukázka designu administrace - uživatelé

List uživatelů je řešen pomocí tabulky. V rámci Bootstrapu byly využity třídy `table`, `table-responsive`, `table-dark` a `table-condensed`.

Každý řádek má svůj podřádek, který je možné si zobrazit pomocí znaku oka ve sloupci akce. Toto řešení student zvolil, kvůli jeho kompaktnosti a přehlednosti a je využíván i na dalších místech v rámci projektu.



Obrázek 5 - ukázka designu administrace - API klíče

## 5.3 Funkce administrace

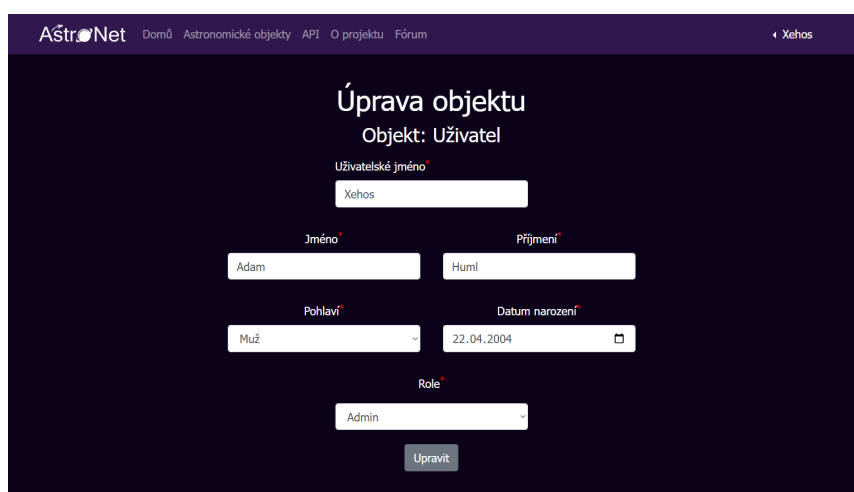
Administrace webu má několik velmi důležitých funkcí, kterými administrátoři web řídí.

### 5.3.1 Funkce správy uživatelů

Ve správě uživatelů se nachází 3 funkce:

1.) Upravit – funkce umožňuje upravit následující atributy libovolného uživatele:

- Uživatelské jméno
- Jméno
- Příjmení
- Pohlaví
- Datum narození
- Role (Admin/Uživatel)



Obrázek 6 - administrace - ukázka úpravy uživatele

2.) Smazat – funkce umožňuje smazat libovolného uživatele (poté co akci potvrdí)  
Řádek uživatele je v tomto případě kompletně vymazán z databáze.

3.) Resetovat heslo – funkce vyžádá změnu hesla uživatele při jeho příštím přihlášení (tento způsob resetu hesla je řešen příznakem v databázi a přejímá inspiraci ze správy uživatelů v Microsoft® Windows)

Název	Typ	Nulový	Výchozí
password_reset	int(11)	Ne	0

Tabulka 2 - MySQL sloupec password\_reset (reset hesla)

### 5.3.2 Funkce správy API klíčů

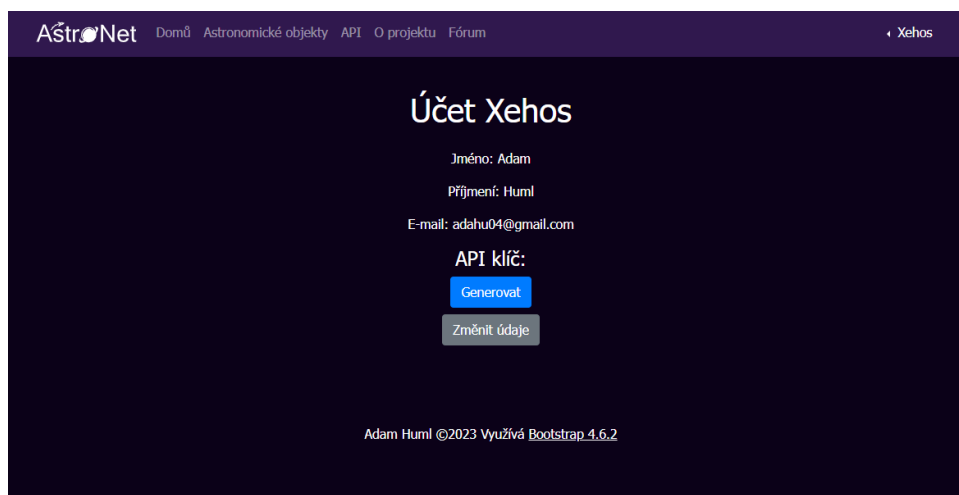
- 1) Ve správě API klíčů se nachází dvě funkce
- 2) Blokovat / Povolit API klíč daného uživatele
- 3) Natavit denní kvótu requestů daného uživatele

## 6. Správa účtu uživatele

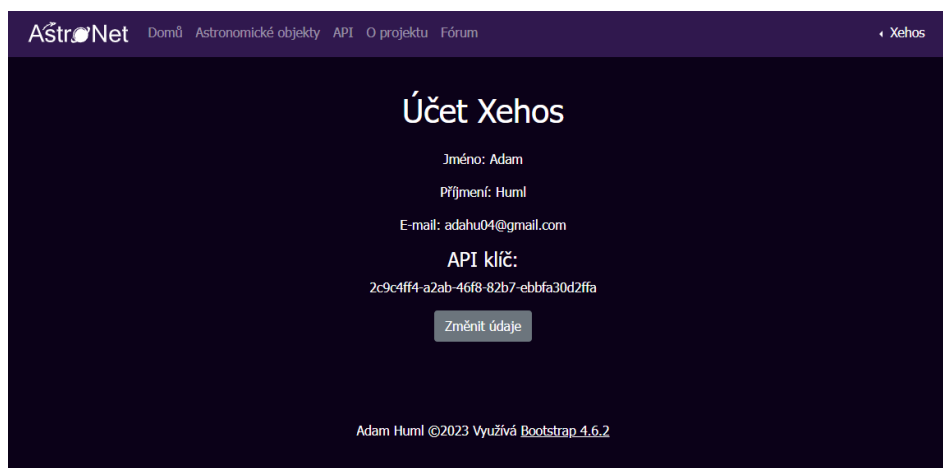
Každý uživatel má možnost spravovat svůj účet prostřednictvím stránky účtu. Tato stránka má jednak informativní funkci, kde může uživatel zkontrolovat pravdivost zadaných údajů, tak i funkci „Změnit údaje“, pomocí které může uživatel upravit následující údaje o svém účtu:

- Jméno
- Příjmení
- E-mail

Zároveň zde může uživatel vygenerovat svůj API klíč (každý uživatel má jen jeden svůj API klíč, který se vygeneruje právě jednou).



Obrázek 7 - účet - ukázka bez vygenerovaného API klíče



Obrázek 8 - účet - ukázka s vygenerovaným API klíčem

## 6.1 Generování API klíčů

Pro API klíče student využil klíče generované v souladu s UUID (Universally unique identifier – Univerzální unikátní identifikátor) verze 4 (RFC 4122).

**Univerzální unikátní identifikátor** je 128bitové číslo používané k identifikaci informací v počítačových systémech. Když jsou UUID generovány podle standardních metod, jsou pro praktické účely jedinečné. Jejich jedinečnost nezávisí na ústředním registračním orgánu, nebo koordinaci mezi stranami, které je vytvářejí. I když není nulová pravděpodobnost, že UUID může existovat duplicitně, je taková možnost dostatečně blízka nule, aby byla zanedbatelná.<sup>[7]</sup>

Pro vygenerování API klíčů v souladu s UUID 4 formátem využil student předem připravenou funkci, která využívá generátor náhodných bajtů k vytvoření základu klíče a poté provádí úpravu těchto bajtů do formátu UUID 4.

(tato funkce pochází z webu <https://www.uuidgenerator.net> a její citace se nachází níže na stránce.

## 1. Roll-Your-Own UUID Generation Function in PHP

This small helper function generates RFC 4122 compliant Version 4 UUIDs.

```
1 function guidv4($data = null) {
2     // Generate 16 bytes (128 bits) of random data or use the data passed into the function.
3     $data = $data ?? random_bytes(16);
4     assert(strlen($data) == 16);
5
6     // Set version to 0100
7     $data[6] = chr(ord($data[6]) & 0x0f | 0x40);
8     // Set bits 6-7 to 10
9     $data[8] = chr(ord($data[8]) & 0x3f | 0x80);
10
11     // Output the 36 character UUID.
12     return vsprintf('%s%s-%s-%s-%s-%s', str_split(bin2hex($data), 4));
13 }
```

*Ukázka kódu 8 - funkce generující UUID 4 API klíče <sup>[8]</sup>*

---

<sup>7</sup>SMUTO, *Univerzální unikátní identifikátor*. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023 [cit. 2023-03-05]. Dostupné z: [https://cs.wikipedia.org/wiki/Univerzální\\_unikátní\\_identifikátor](https://cs.wikipedia.org/wiki/Univerzální_unikátní_identifikátor)

<sup>8</sup>*Generate a UUID in PHP. UUID Generator* [online]. TransparenTech [cit. 2023-03-06]. Dostupné z: <https://www.uuidgenerator.net/dev-corner/php>

## 7. API

Registrovaní uživatelé mají možnost využít RESTful (Representational State Transfer) API dostupné pomocí protokolu HTTP(S) pro čerpání dat z aplikace.

RESTful API v tomto případě umožňuje poskytování dat na základě povolených zdrojů a metod poskytovaných protokolem HTTP. Takových metod je k dispozici hned několik, nicméně tato práce závisí primárně na metodách GET a POST.<sup>[9]</sup>

API běží v Pythonu a využívá k chodu HTTP server Uvicorn ze stejnojmenného modulu a modul FastAPI, který umožňuje snadnou implementaci API.

Přistupuje k databázi naprosto nezávisle od webové aplikace, nicméně interpretuje totožná data ve formátu JSON.

Účel API je poskytnout data shromážděná v aplikaci Astronet dalším vývojářům a umožnit jim provést implementaci do vlastních aplikací a programů.

API je v zájmu zachování všestrannosti jeho využití psáno v anglickém jazyce.

### 7.1 Třída DBHandler

Třída DBHandler umožňuje snadnou práci s databází MySQL. K jejímu připojení při tom využívá modul `mysql.connector`.

Užívá se ke všem přístupům k API

```
class DBHandler:

    def __init__(self, **kwargs):
        try:
            self.db = mysql.connector.connect(
                host=os.environ["MYSQL_HOSTNAME"],
                user=os.environ["MYSQL_USER"],
                password=os.environ["MYSQL_PASSWORD"],
                database=os.environ["MYSQL_DB"]
            )
        except KeyError:
            raise Exception("DB Connection failed! Check credentials!")

        self.cursor = self.db.cursor()
```

*Ukázka kódu 9 - třída DBHandler - inicializace třídy DBHandler pomocí proměnných prostředí*

Třída DBHandler je iniciována na základě globálních proměnných prostředí zapsaných v souboru `.env` v kořenové složce API ve formátu YAML. Tento postup pochází z containerizace a někdy se tento `.env` soubor nazývá také manifest.

---

<sup>9</sup>HANÁK, Drahomír, ČÁPKA, David, ed. *Stopařův průvodce REST API*. ITnetwork.cz [online]. Česká republika: ITnetwork.cz, 2013 [cit. 2023-03-06]. Dostupné z:

<https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api/>

<sup>10</sup>Environment manifest (`env.yaml`). Amazon AWS [online]. Seattle, Washington, USA: AWS, 2017 [cit. 2023-03-06]. Dostupné z:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-cfg-manifest.html>



Dále si student vytvořil v třídě DBHandler funkci gather\_data, která vrátí list objektů a umožňuje mu snadný výběr potřebných dat z databáze na základě zadaných atributů. Tzn. formátuje SQL příkaz SELECT.

```
def gather_data(self, table: str, limit: int=-1, attrs: dict={}) -> List:
    objects = []
    if attrs == {}:
        if limit > -1:
            self.cursor.execute("SELECT * FROM {} LIMIT {}".format(table, limit))
        else:
            self.cursor.execute("SELECT * FROM {}".format(table))
    else:
        if limit > -1:
            self.cursor.execute("SELECT * FROM {} WHERE {} = '{}' LIMIT {}".format(table,
List(attrs.keys())[0], attrs[List(attrs.keys())[0]], limit))
        else:
            self.cursor.execute("SELECT * FROM {} WHERE {} = '{}'".format(table,
List(attrs.keys())[0], attrs[List(attrs.keys())[0]]))

    myresult = self.cursor.fetchall()
    field_names = [i[0] for i in self.cursor.description]
    for x in range(len(myresult)):
        res_dict = {}
        for z in range(len(myresult[x])):
            res_dict[field_names[z]] = myresult[x][z]
        objects.append(res_dict)
    self.db.commit()

    return objects
```

*Ukázka kódu 10 - třída DBHandler - funkce gather\_data umožňující výběr dat z databáze*

## 7.2 Nastavení resources API – api.py

V souboru api.py jsou zapsána nastavení jednotlivých zdrojů RESTful API. Tzn. jeho jednotlivých přístupových bodů. Pro uživatele jsou dostupné následující zdroje:

- / - defaultní zdroj – vypíše základní informace o účelu API
- /ssplanets – zdroj pro získání dat o planetách sluneční soustavy
- /satellites – zdroj pro získání dat o satelitech
- /stars – zdroj pro získání dat o hvězdách
- /exoplanets – zdroj pro získání dat o exoplanetách

Dále jsou pak definovány další dva pomocné zdroje přístupné pouze ze serveru samotného. Ty umožňují přidání / smazání uživatele ve fóru. Tento způsob je dále detailně popsán v [kapitole 10](#). Za tímto účelem jsou využívány následující zdroje:

- /addforumuser – přidání uživatele fóra
- /delforumuser - smazání uživatele fóra

## 7.3 Funkce check\_api\_key(api\_key)

Funkce check\_api\_key s atributem api\_key (api klíč) umožňuje kontrolu API klíče před návratem vyžadovaných dat. Zároveň funkce zjišťuje, zdali uživatel nepřekročil jemu přidělenou denní kvótu requestů na API server.

```
def check_api_key(api_key) -> bool:
    api_key_list = db.gather_data("astronet_api_keys",1,{"api_key":api_key})
    print(api_key_list)
    if len(api_key_list)>0:
        api_key = api_key_list[0]
        if api_key["requests_today"] < api_key["requests_quota"] and api_key["revoked"] == 0:
            db.add_number("astronet_api_keys",
                {
                    "number_col":"requests_today",
                    "number":1,
                    "selector_col":"api_key",
                    "selector":api_key["api_key"]
                }
            )
        return True
    return False
```

*Ukázka kódu 11 - api.py - ukázka funkce check\_api\_key*

## 7.4 Oddělení API od aplikace

API běží na samostatném HTTP serveru, který je poskytován modulem Uvicorn pro Python. Je také nasměrován na samostatnou doménu <https://api.astronet.fun>.

```
server{
    server_name api.astronet.fun;
    location / {
        proxy_pass http://localhost:2426;
        proxy_set_header X-Forwarded-Proto https;
        #client_max_body_size 20M;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/api.astronet.fun/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/api.astronet.fun/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

*Obrázek 9 - ukázka nastavení NGINX reverzní proxy pro API*

## 7.5 Výstup z API

API navrácí uživateli prostřednictvím protokolu HTTP(S) soubor JSON obsahující žádaná data. Příklad výstupu z API je přiložen na konci této práce.

JSON (JavaScript Object Notation) je formát pro výměnu dat. Vyznačuje se tím, že je pro člověka snadné ho číst, či psát. Pro stroje je pak snadné dokumenty v tomto formátu analyzovat a generovat. Je založen na podmnožině standardu JavaScript Programming Language Standard ECMA-262 3rd Edition. JSON je textový na jazyce zcela nezávislý formát, ale používá konvence, které jsou známé programátorům z rodiny jazyků C, včetně C, C++, C#, Java, JavaScript, Perl, Python a mnoho dalších. Tyto vlastnosti dělají z JSONu ideální formát pro výměnu dat.<sup>[11]</sup>

---

<sup>11</sup>Úvod do JSON. *Json.org* [online]. [cit. 2023-03-14]. Dostupné z: <https://www.json.org/json-cz.html>

## 8. Astronomické objekty

Další částí práce je seznam vesmírných těles nacházející se pod položkou hlavního menu „Astronomické objekty“.

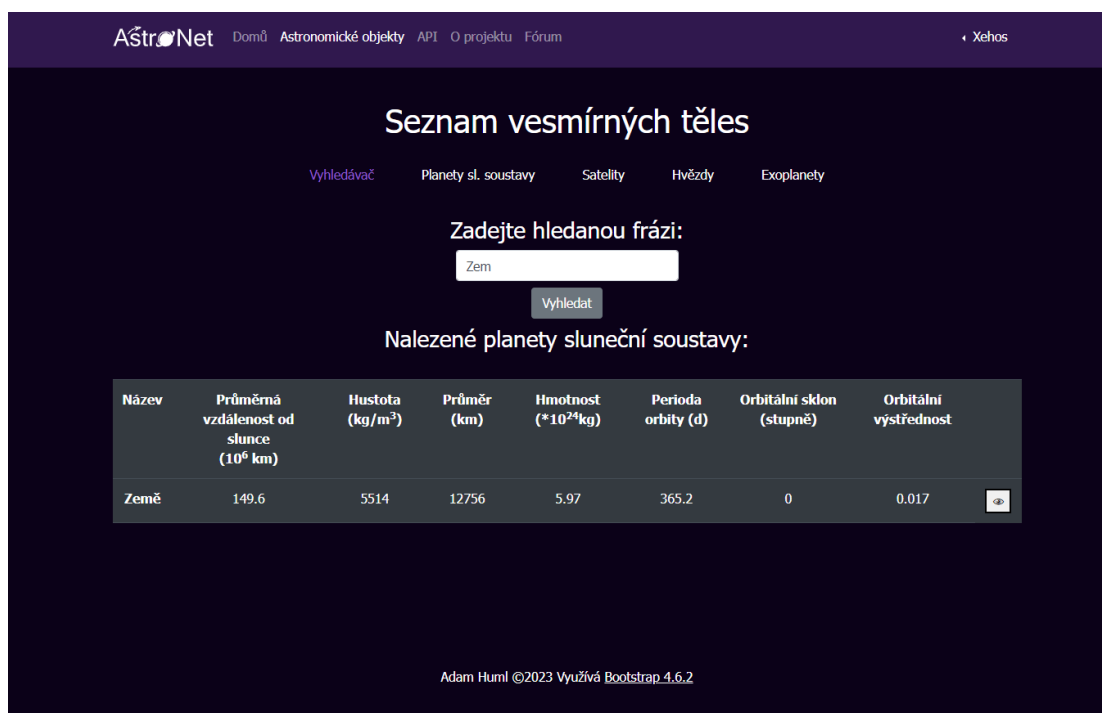
Všichni přihlášení i nepřihlášení uživatelé aplikace mají možnost prohlížet si vesmírná tělesa celkem ze 4 kategorií:

- Planety sluneční soustavy
- Satelity – přirozené i uměle vytvořené
- Hvězdy
- Exoplanety


Každá kategorie má svůj vlastní formát tabulky s údaji. Ta se skládá z hlavního řádku a skrytého řádku, který lze zobrazit kliknutím na tlačítko se symbolem oka po pravé straně hlavního řádku tabulky. Jednotky údajů jsou popsány v hlavičce tabulky.

### 8.1 Vyhledávač

**Přihlášeným uživatelům** je umožněno vyhledávat v databázi podle názvu astronomického objektu. Minimální počet znaků pro vyhledávání je 3. V případě, že vyhledávač najde shodu ve více než jedné tabulce, jsou tabulky vypsány pod sebou.



The screenshot shows the AštrONet website with a search bar containing 'Zem' and a 'Vyhledat' button. Below the search bar, the results are displayed under the heading 'Nalezené planety sluneční soustavy:'. The results table has the following data:

Název	Průměrná vzdálenost od slunce (10 <sup>6</sup> km)	Hustota (kg/m <sup>3</sup> )	Průměr (km)	Hmotnost (*10 <sup>24</sup> kg)	Perioda orbity (d)	Orbitální sklon (stupně)	Orbitální výstřednost	
Země	149.6	5514	12756	5.97	365.2	0	0.017	

At the bottom of the page, it says 'Adam Huml ©2023 Využívá Bootstrap 4.6.2'.

Obrázek 10 - objekty - design vyhledávače

Vyhledávač používá SQL příkaz SELECT s operátorem LIKE, který umožňuje vyhledávat na základě zadaného řetězce. Příkaz vypadá zhruba takto:

```
SELECT * FROM `tabulka` WHERE sloupec LIKE %řetězec%;
```

% v příkazu vyjadřují místa, kde je možné doplnit jakkoliv dlouhý řetězec. Tzn. „nalezneme všechny řetězce, obsahující řetězec“.

Vyhledávání není *case sensitive* (nebere ohled na velká a malá písmena).

## 8.2 Planety sluneční soustavy

Tabulka obsahuje seznam planet sluneční soustavy. Data jsou uložena v MySQL tabulce *astronet\_ssplanets* (solar system planets → planety sluneční soustavy).

Struktura tabulky:





### Hlavní řádek

Hlavička tabulky	Název*	Průměrná vzdálenost od Slunce*	Hustota*	Průměr*	Hmotnost*	Perioda orbity*	Orbitální sklon*	Orbitální výstřednost*
Jednotka	—	km * 10 <sup>6</sup>	kg/m <sup>3</sup>	km	kg * 10 <sup>24</sup>	den	stupně	—
MySQL sloupec	name	distance_from_sun	density	diameter	mass	orbital_period	inclination	eccentricity
Význam	Název planety	Průměrná vzdálenost planety od Slunce	Hustota planety (m/V)	Průměr planety	Hmotnost planety	Orbita obíhaná planetou kolem Slunce	Míra sklonu oběžné dráhy kolem kosmického tělesa.	Bezrozměrná veličina, určující odchylku orbity planety od dokonalé kružnice

### Skrýty řádek

Hlavička tabulky	Model*	Popis*	Odkaz na API
MySQL sloupec	3d_model	description	—
Význam	3D model planety	Popis planety	Odkaz na API vypisující data o konkrétní planetě (pouze pro přihlášené) – API klíč je dosazen.

\* atribut je povinný a je nutné jej vyplnit při vytváření planety

Název	Průměrná vzdálenost od slunce (10 <sup>6</sup> km)	Hustota (kg/m <sup>3</sup> )	Průměr (km)	Hmotnost (*10 <sup>24</sup> kg)	Perioda orbity (d)	Orbitální sklon (stupně)	Orbitální výstřednost	
Merkur	57.9	5429	4879	0.33	88	7	0.206	
Venuše	108.2	5243	12104	4.87	224.7	3.4	0.007	
Země	149.6	5514	12756	5.97	365.2	0	0.017	
Model		Popis					API	
		<p>Země – největší z planet zemského typu. Je jedinou planetou v celém vesmíru, o které víme, že na ní existuje život. Má dostatečně hustou atmosféru, dostatek kapalné vody v povrchových oceánech. Kolem Země obíhá jediný měsíc s vázanou rotací. Při pozorování Země z kosmu vidíme hlavně modrou barvu oceánů. 70 % povrchu Země je pokryto oceány, 30 % tvoří kontinenty. Země sestává z těchto vrstev: jádro, plášť, kůra, troposféra, stratosféra, mezosféra, termosféra. Plášť a kůra jsou odděleny tzv. Mohorovičičovým rozhraním. Kůra se posouvá a „plave“ na polotekutém plášti. Teplota v centru Země je 5 100 °C, tlak 360 GPa. Magnetické pole Země má přibližně dipólový charakter, je deformováno slunečním větrem do typického tvaru.</p>					<a href="#">API</a>	
Admin		<a href="#">Upravit</a> <a href="#">Smazat</a>						

Obrázek 11 - Objekty - design záznamu planety sluneční soustavy

## 8.3 Satelity

### Hlavní řádek

Hlavička tabulky	Název*	Planeta*	Vzdálenost od planety*	Průměr*	Hmotnost*	Perioda orbity*	Orbitální sklon*	Orbitální výstřednost*
Jednotka	—	—	km	km	kg * 10 <sup>24</sup>	den	stupně	—
MySQL sloupec	name	planet_id	distance_from_planet	diameter	mass	orbital_period	inclination	eccentricity
Význam	Název planety	Název planety obíhané satelitem. <i>Převod přes ID z tabulky astronetsplanets</i>	Vzdálenost satelitu od planety	Průměr satelitu (pouze pro přirozené satelity)	Hmotnost planety	Orbita obíhaná satelitem kolem planety	Míra sklonu oběžné dráhy kolem kosmického tělesa	Bezrozměrná veličina, určující, odchylku orbity satelitu od dokonalé kružnice

### Skrytý řádek

Hlavička tabulky	Model	Popis*	Odkaz na API
MySQL sloupec	3d_model	description	—
Význam	3D model satelitu	Popis satelitu	Odkaz na API vypisující data o konkrétním satelitu (pouze pro přihlášené) – API klíč je dosazen.

\* atribut je povinný a je nutné jej vyplnit při vytváření satelitu

Envisat	Země	800	8950	357900	98.52	0.00017	
Spoutnik 1	Země	1000	5.84E-5	88.2	65	0.7	
GPS Block II	Země	20200	1350	71400	55	0.0008	
Hubbleův vesmírný dalekohled	Země	569	11400	60250	28.5	0.0004	

Model	Popis	API
	Hubbleův vesmírný dalekohled je dalekohled, který na oběžnou dráhu Země do výše 600 kilometrů vynesl v roce 1990 při letu STS-31 americký raketoplán Discovery. Současně obíhá Zemí ve výšce asi 569 km. Dalekohled předává na Zemi obrazy vesmíru neovlivněné zemskou atmosférou.	<div>API</div>
Admin	<div>Upravit</div> <div>Smazat</div>	

Obrázek 12 - Objekty - design záznamu satelitu

## 8.4 Hvězdy


### Hlavní řádek

Hlavička tabulky	Název*	Vzdálenost od Země*	Vzdálenost od Slunce*	Hvězdná velikost*	Barva*	Světelnost*	Hmotnost*
Jednotka	—	AU	AU	mag	—	poměr ku Slunci	poměr ku Slunci
MySQL sloupec	name	planet_id	distance_from_planet	diameter	color	luminosity	mass
Význam	Název hvězdy	Vzdálenost hvězdy od Země	Vzdálenost hvězdy od Slunce	Zdánlivá magnituda hvězdy - udává jasnost	Barva hvězdy v češtině	Světelnost hvězdy	Hmotnost hvězdy

### Skrytý řádek

Hlavička tabulky	Popis*	Odkaz na API
MySQL sloupec	description	—
Význam	Popis hvězdy	Odkaz na API vypisující data o konkrétní hvězdě (pouze pro přihlášené) – API klíč je dosazen.

\* atribut je povinný a je nutné jej vyplnit při vytváření hvězdy

Název	Vzdálenost od Země (AU)	Vzdálenost od Slunce (AU)	Hvězdná velikost	Barva	Světelnost (ku Slunci)	Hmotnost (ku Slunci)	
Slunce	1	0	-26.74	žlutá	1	1	
Proxima Centauri	4.24	268551	11.05	červená	0.00156	0.123	
Popis						API	
Proxima Centauri je nejbližší hvězda k naší sluneční soustavě. Její teplota je nižší než teplota Slunce, ale její hmotnost je pouze 12% hmotnosti Slunce.						<a href="#">API</a>	
Admin						<a href="#">Upravit</a> <a href="#">Smazat</a>	
Sirius	8.6	8.6	-1.46	modrá	25.4	2.02	
Alpha Centauri A	4.37	271297	0.01	bílá	1.519	1.1	

Obrázek 13 - Objekty - design záznamu hvězdy

## 8.5 Exoplanety

### Hlavní řádek

Hlavička tabulky	Název*	Mateřská hvězda*	Vzdálenost od mateřské hvězdy*	Hmotnost*	Orbitální sklon*	Orbitální výstřednost*	Potenciálně obyvatelná*
Jednotka	—	—	AU	$kg * 10^{24}$	stupně	poměr ku Slunci	ANO/NE
MySQL sloupec	name	planet_id	distance_from_planet	diameter	inclination	eccentricity	potentially_habitable
Význam	Název hvězdy	Vzdálenost hvězdy od Země	Vzdálenost hvězdy od Slunce	Zdánlivá magnituda hvězdy - udává jasnost	Míra sklonu oběžné dráhy kolem kosmického tělesa	Bezrozměrná veličina, určující, odchylku orbity exoplanety od dokonalé kružnice	Určuje zdali je exoplaneta potenciálně obyvatelná lidmi

### Skrytý řádek

Hlavička tabulky	Model	Popis*	Odkaz na API
MySQL sloupec	3d_model	description	—
Význam	3D model exoplanety	Popis exoplanety	Odkaz na API vypisující data o konkrétní exoplanetě (pouze pro přihlášené) – API klíč je dosazen.

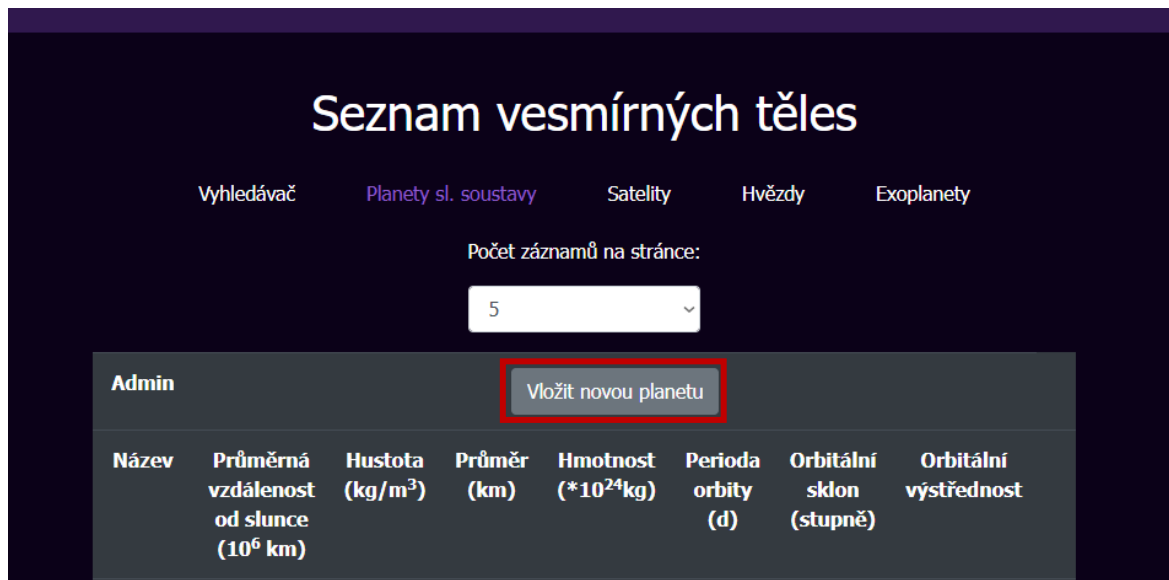
Název	Mateřská hvězda	Vzdálenost od mateřské hvězdy (AU)	Hmotnost (* $10^{24}$ kg)	Orbitální sklon	Orbitální výstřednost (stupně)	Potenciálně obyvatelná	
Proxima b	Proxima Centauri	0.05	0.004	55	0.35	ANO	
Model		Popis				API	
		Proxima b je exoplaneta, která se nachází v nejbližší známé hvězdě Proxima Centauri. Jeho hmotnost je podobná Zemi, a proto by mohla mít pevný povrch a atmosféru. Někteří vědci se domnívají, že by mohla být obyvatelná, i když byla objevena v oblasti, kde je vysoká radiace a povrch by mohl být rozpálený.				<a href="#">API</a>	
Admin		<div>Upravit</div> <div>Smazat</div>					
Kepler-438b	Kepler-438	0.1648	0.12	34.8	0.26	ANO	

Obrázek 14 - Objekty - design záznamu exoplanety

## 9. Manipulace s daty

### 9.1 Vkládání nových objektů

Administrátoři mají možnost vkládat nové objekty pomocí tlačítka sloužícího pro vložení nového objektu, které se nachází nad hlavičkou příslušné tabulky objektů.




Obrázek 15 - Manipulace s daty - ukázka tlačítka pro vložení nového objektu (planety)

Obrázek 16 - Manipulace s daty - ukázka designu formuláře pro vložení nového objektu (planety)



## 9.2 Úprava objektů

Administrátoři mají možnost upravovat záznamy objektů. Činí tak prostřednictvím ovládacího panelu „Admin“, který se jim zobrazí ve skrytém řádku tabulky.

Merkur	57.9	5429	4879	0.33	88	7	0.206	
Model	Popis						API	
	Merkur - planeta nejbližší Slunci. Je to skalnatá planeta, posetá krátery podobně jako náš Měsíc. Jde o nejmenší planetu vůbec. Je téměř bez atmosféry. Teplota povrchu tohoto tělesa kolísá mezi -180 °C a 430 °C. Merkur se otočí kolem vlastní osy jednou za 59 našich dnů. Jeho doba oběhu kolem Slunce trvá 88 dnů. Jde o příklad vázané rotace (spinorbitální interakce) v poměru 2:3 způsobené slapovými silami. Dráha Merkuru kolem Slunce je protáhlá elipsa, která se stáčí vlivem přítomnosti ostatních planet. Malá část stáčení perihelia dráhy (43" za století) je způsobena efekty obecné relativity.						<a href="#">API</a>	
Admin	<div><a href="#">Upravit</a></div>						<div><a href="#">Smazat</a></div>	

Obrázek 17 - Úprava objektů - ukázka tlačítka pro úpravu objektu (planety)

AstroNet

Domů Astronomické objekty API O projektu Fórum

Xehos

## Úprava objektu

Objekt: Merkur

Název\*

Merkur

Průměrná vzdálenost od slunce (10<sup>6</sup> km)\*

57.9

Hustota (kg/m<sup>3</sup>)\*

5429

Průměr (km)\*

4879

Hmotnost (\*10<sup>24</sup>kg)\*

0.33

Perioda orbity (d)\*

88

Orbitální sklon (stupně)\*

7

Orbitální výstřednost\*

0.206

Popis\*

Merkur - planeta nejbližší Slunci. Je to skalnatá planeta, posetá krátery podobně jako náš Měsíc. Jde o nejmenší planetu vůbec. Je téměř bez atmosféry. Teplota povrchu

Jméno 3D modelu (nepovinné)

Mercury\_1\_4878.glb

Pořadí ve sluneční soustavě\*

1

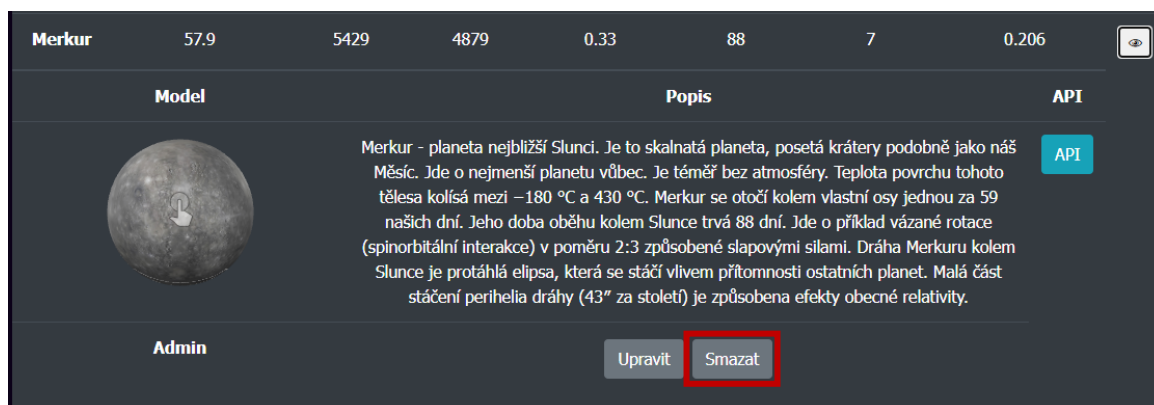
Upravit

Adam Huml ©2023 Využívá Bootstrap 4.6.2

Obrázek 18 - Úprava objektů - design úpravy objektu (planeta sl. soustavy - Merkur)

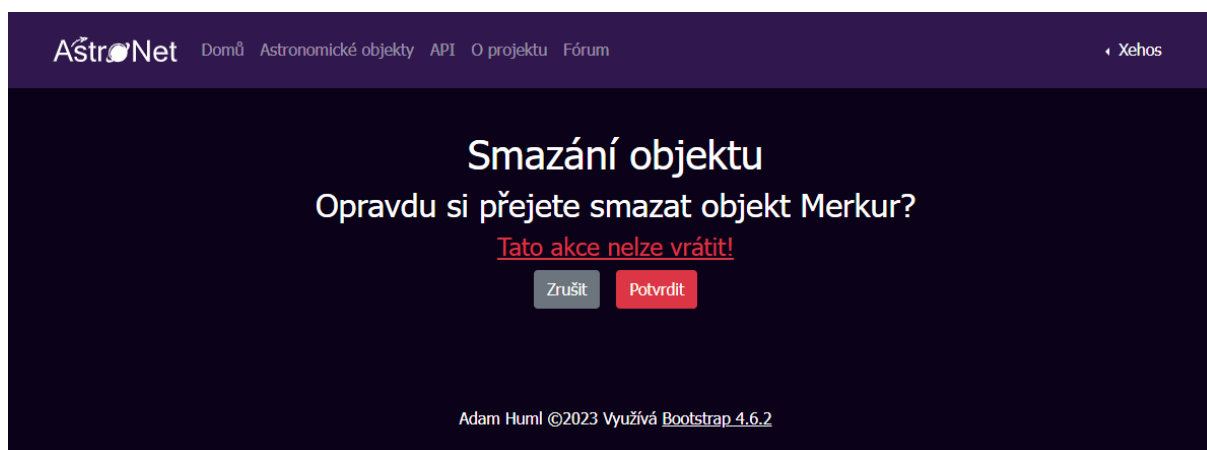
### 9.3 Mazání objektů

Administrátoři také mohou jednotlivé záznamy mazat. K tomu stejně, jako u úpravy objektů využijí ovládacího panel „Admin“, který se jim zobrazí ve skrytém řádku tabulky.



Obrázek 19 - Mazání objektů - ukázka tlačítka pro smazání objektu (planety)

Smazání objektu vyžaduje potvrzení. Administrátor je bezprostředně po stisknutí tlačítka smazání přesměrován na stránku, kde musí akci potvrdit. Pokud tak neučiní (případně stiskne-li tlačítko „Zrušit“) vymazání záznamu se neprovede.



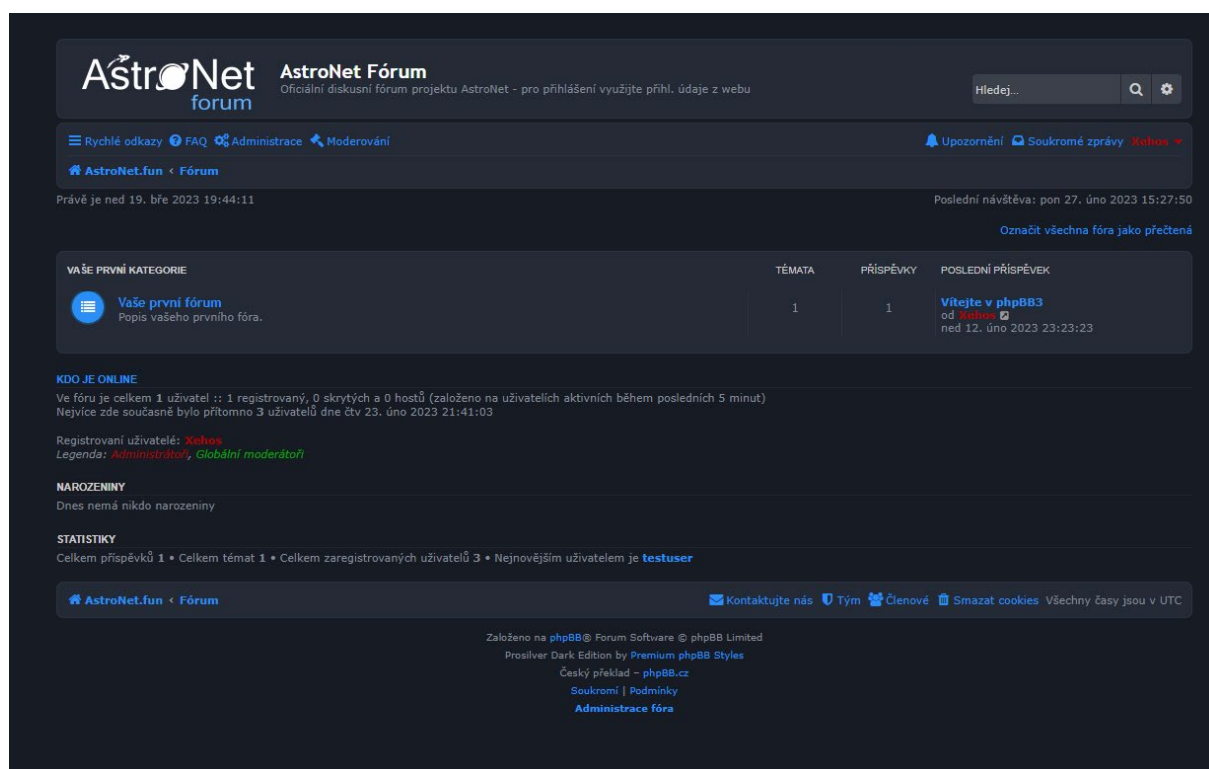
Obrázek 20 - Mazání objektů - ukázka potvrzení smazání objektu

## 10. Uživatelské diskusní fórum

Jak již bylo uvedeno v úvodu práce, její součástí je implementace uživatelského diskusního fóra běžícího na platformě phpBB®.

K využití externího opensource kódu tohoto fóra student přistoupil z důvodu, že by případná implementace vlastního fóra byla dosti časově náročná, a možná by dokonce i překonala rozsah práce.

phpBB® je platforma licencována pod licencí GNU (General Public License). Tj. licence pro svobodný software, původně napsaná Richardem Stallmanem pro projekt GNU (GNU's Not Unix!). GPL je nejpopulárnějším příkladem tzv. copyleftové licence, která vyžaduje, aby byla odvozená díla dostupná pod toutéž licencí.<sup>[12]</sup>



Obrázek 21 - Uživatelské diskusní fórum - Ukázka fóra na platformě phpBB® po instalaci

<sup>12</sup>STALLMAN, Richard. GNU GENERAL PUBLIC LICENSE. *GNU Operating System* [online]. Boston, Massachusetts, USA: Free Software Foundation, 2007 [cit. 2023-03-19]. Dostupné z: <https://www.gnu.org/licenses/gpl-3.0.html>

## 10.1 Implementace fóra do aplikace

Pro plnou implementaci fóra do aplikace AstroNet bylo třeba, aby se ve fóru využívalo k přihlášení stejných registračních údajů, jako do samotné aplikace. Toho student dosáhl pomocí API, kdy pomocí cli rozhraní phpBB volá funkce pro vytvoření/smazání uživatele při jeho vytvoření/smazání v aplikaci. Tento API resource je z důvodu bezpečnosti limitován na IP localhost (127.0.0.1).

```
@api_router.get("/addforumuser/", tags=["App", "Admin"])
def add_user(username:str, password:str, mail:str, request: Request):

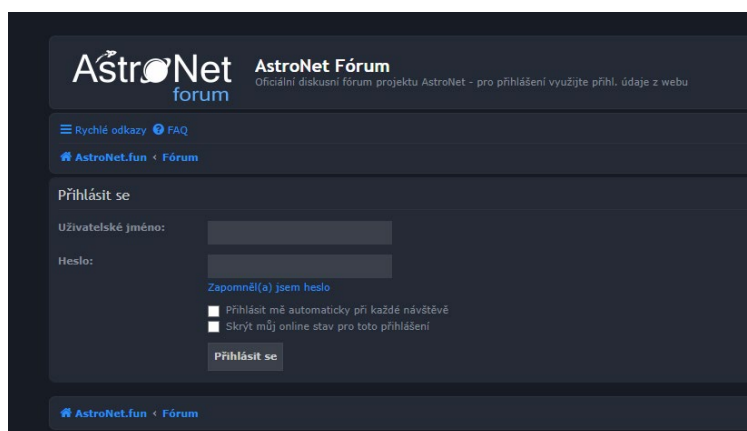
    client_host = str(request.client.host)
    if client_host != "127.0.0.1":
        return {
            "loc": [
                "permissions"
            ],
            "request": "failed",
            "error_message": "Access denied",
            "type": "access.denied"
        }
    else:
        os.system("php forum/bin/phpbbcli.php -U {} -P {} -E {}".format(username, password, mail))
        return {
            "status": "ok",
            "user": {
                "username": username,
                "password": password,
                "e-mail": mail
            }
        }
```

*Ukázka kódu 12 - Uživatelské diskusní fórum - ukázka implementace přidání nového uživatele do fóra*

Po registraci do aplikace, se tedy uživatel může do fóra přihlásit běžným způsobem. Samovolné registrace do fóra jsou v administraci phpBB® deaktivované.



*Obrázek 22 - Uživatelské diskusní fórum - deaktivace registrací v phpBB®*



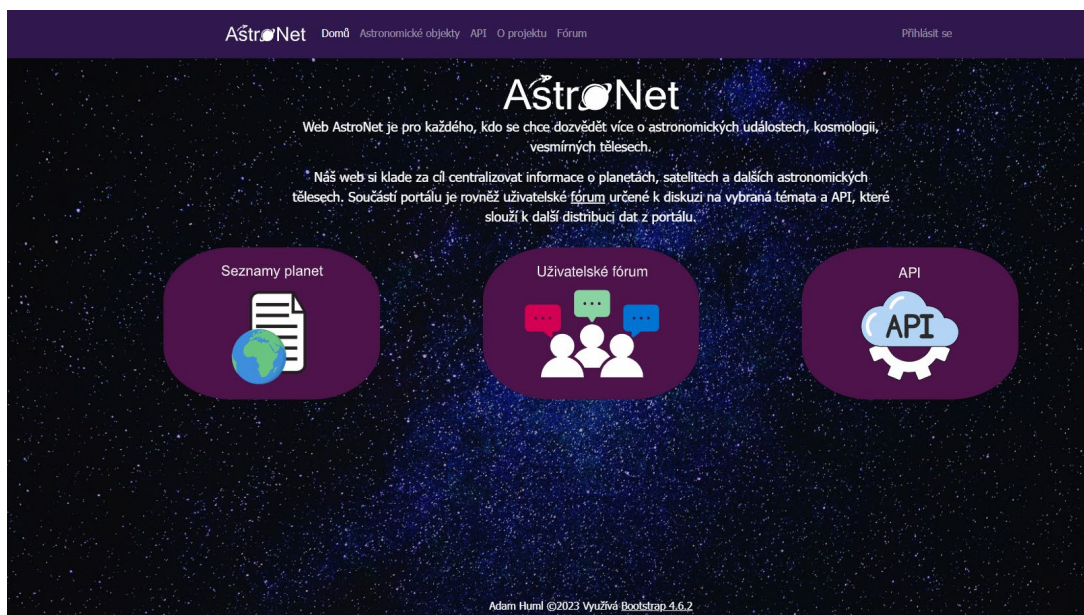
*Obrázek 23 - Uživatelské diskusní fórum - ukázka přihlášení do fóra phpBB®*

## 11. Závěr

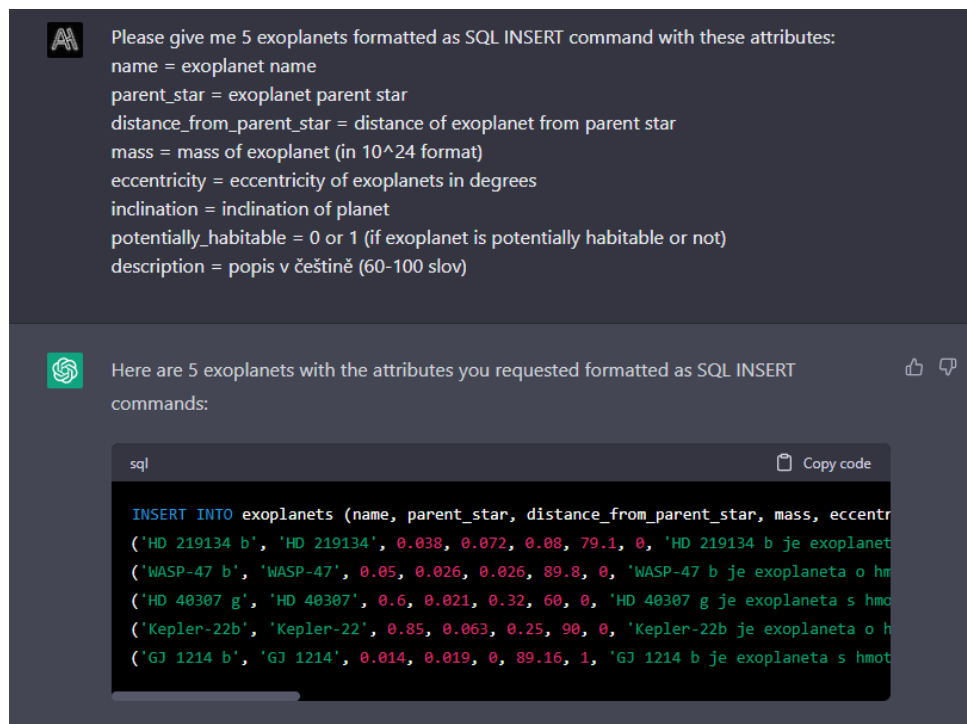
Aplikace AstroNet byla vytvořena především s využitím PHP a Pythonu a slouží k další distribuci zformátovaných astronomických dat. Aplikaci lze využít až již pro vědecké, či studijní účely v rámci práce s danými daty. Výběr dat s aplikace je možné provádět manuálně i strojově (pomocí API).

Součástí je rovněž fórum, které aplikaci doplňuje o prostor k interakci s dalšími uživateli.

Pro účel prezentace, byla aplikace částečně naplněna daty z datového modelu GPT-3 umělé inteligence společnosti OpenAI Inc. Data byla čerpána prostřednictvím služby ChatGPT.



Obrázek 24 - design domovské stránky aplikace



Obrázek 25 - naplnění aplikace daty z modelu umělé inteligence GPT-3 společnosti OpenAI Inc.

## 12. Seznam použitých zdrojů

### 12.1 Literární rešerše

- *Form controls*. Bootstrap [online]. 2022 [cit. 2023-01-21]. Dostupné z: <https://getbootstrap.com/docs/4.6/components/forms/#form-controls>
- *SQL Injection*. Wikipedia [online]. 2010-05-26 [cit. 2023-01-22]. Dostupné z: [https://en.wikipedia.org/wiki/SQL\\_injection?oldid=364265627](https://en.wikipedia.org/wiki/SQL_injection?oldid=364265627)
- KOĐOUSKOVÁ, Barbora. *HTTPS V KOSTCE: CO TO JE, JAK FUNGUJE A JAK NA NĚJ PŘEJÍT*. Rascasone [online]. 2021, 2021-11-17 [cit. 2023-01-22]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-https-http-ssl-tls>
- BAIG, Anas. *What is Session Hijacking and How Do You Prevent It?*. GlobalSign Blog [online]. 2021, 2021-07-09 [cit. 2023-01-22]. Dostupné z: <https://www.globalsign.com/en/blog/session-hijacking-and-how-to-prevent-it>
- MLEJNEK, Pavel, MAKEŠ, Václav, Jan KOVÁŘ BK a Milan KERŠLÁGER, ed. *Let's Encrypt*. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023, 2015-09-26 [cit. 2023-03-06]. Dostupné z: [https://cs.wikipedia.org/wiki/Let%27s\\_Encrypt](https://cs.wikipedia.org/wiki/Let%27s_Encrypt)
- *Webová Administrace: Co to znamená "Webová Administrace"?*. In: *DAMI development* [online]. Česká republika: DAMI development [cit. 2023-03-05]. Dostupné z: <https://www.damidev.com/slovník/webova-administrace>
- SMUTO, *Univerzální unikátní identifikátor*. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023 [cit. 2023-03-05]. Dostupné z: [https://cs.wikipedia.org/wiki/Univerzální\\_unikátní\\_identifikátor](https://cs.wikipedia.org/wiki/Univerzální_unikátní_identifikátor)
- *Generate a UUID in PHP. UUID Generator* [online]. TransparenTech [cit. 2023-03-06]. Dostupné z: <https://www.uuidgenerator.net/dev-corner/php>
- HANÁK, Drahomír, ČÁPKA, David, ed. *Stopařův průvodce REST API*. ITnetwork.cz [online]. Česká republika: ITnetwork.cz, 2013 [cit. 2023-03-06]. Dostupné z: <https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api/>
- *Environment manifest (env.yaml)*. *Amazon AWS* [online]. Seattle, Washington, USA: AWS, 2017 [cit. 2023-03-06]. Dostupné z: <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-cfg-manifest.html>
- *Úvod do JSON*. *Json.org* [online]. [cit. 2023-03-14]. Dostupné z: <https://www.json.org/json-cz.html>
- STALLMAN, Richard. *GNU GENERAL PUBLIC LICENSE. GNU Operating System* [online]. Boston, Massachusetts, USA: Free Software Foundation, 2007 [cit. 2023-03-19]. Dostupné z: <https://www.gnu.org/licenses/gpl-3.0.html>

## 12.2 Použito při tvorbě práce

- Imfmaier. Cities-json. *Github* [online]. Github, 2019 [cit. 2023-03-19]. Dostupné z: <https://github.com/Imfmaier/cities-json>
- BOINTON, Marcus. PHPMailer: A full-featured email creation and transfer class for PHP. *Github* [online]. Github, 2014 [cit. 2023-03-20]. Dostupné z: <https://github.com/PHPMailer/PHPMailer>

## 12.3 Použito při plnění práce daty

- WILLIAMS, Dr. David R. Planetary Fact Sheet - Metric. *NASA Space Science Data Coordinated Archive [NSSDC]* [online]. NASA Goddard Space Flight Center: NASA, 2023 [cit. 2023-03-20]. Dostupné z: <https://nssdc.gsfc.nasa.gov/planetary/factsheet/>
- ARNETT, Bill. The Nine Planets: An overview of the history, mythology and current scientific knowledge of the planets, moons and other objects in our solar system. *The Nine Planets* [online]. 1994 [cit. 2023-03-20]. Dostupné z: <https://nineplanets.org>
- *Aldebaran* [online]. Česká republika: FEL ČVUT, 2002 [cit. 2023-03-19]. Dostupné z: <https://www.aldebaran.cz>
- *Chat GPT* [online]. San Francisco, USA: OpenAI Inc., 2022 [cit. 2023-03-20]. Dostupné z: <https://chat.openai.com/>

*Pozn.: Chat GPT jako model umělé inteligence NELZE považovat za spolehlivý zdroj faktických informací. V rámci práce byl využit pouze za účelem generování dat pro prezentaci jejího reálného využití.*

## 12.4 3D modely

- DAVIS, Phillips, Lonnie SHEKHTMAN a Jay THOMPSON. NASA Solar System Exploration Resources: Our Galactic Neighbourhood. *NASA Solar System Exploration* [online]. Washington, D.C.: NASA, 2023 [cit. 2023-03-20]. Dostupné z: <https://solarsystem.nasa.gov>